

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
Кафедра автоматизованих систем обробки інформації і управління

УДК: 004

«До захисту допущено»
В.о. завідувача кафедри
О.А.Павлов
(підпис) (ініціали, прізвище)
“ ” 2019 р.

Дипломний проект
на здобуття ступеня бакалавра

з напрямку підготовки 6.050101 «Комп'ютерні науки»

на тему: «Інформаційна підтримка вибору наукового керівника кваліфікаційної роботи»

Виконав:

студент 4 курсу, групи ІС-51

Как Самір Рахулович
(прізвище, ім'я, по батькові)

(підпис)

Керівник

доц., к.т.н., доц. Тєлишева Т.О.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

**Консультант з
графічної
документації**

доц., к.т.н., доц. Тєлишева Т.О.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Рецензент

доц., к.т.н., доц. Ткач. М.М.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Засвідчую, що у цьому дипломному проекті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент Как С.Р.

(підпис)

Київ – 2019 року

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка дипломного проекту складається з п'яти розділів, містить 20 рисунків, 12 таблиць, 1 додаток, 10 джерел.

Дипломна робота присвячена створенню інформаційної підтримки вибору наукового керівника кваліфікаційної роботи.

Основними цілями розробки системи, на якій базується інформаційна підтримка є:

- усвідомлений та мотивований пошук керівника;
- впорядкування комфортного робочого процесу для студента і керівника;
- підвищення якості кваліфікаційних робіт внаслідок сумісної мотивованої роботи та зацікавленості обох сторін у подальшому розвитку співпраці;
- створення розгорнутого списку потенціальних керівників.

Для досягнення поставлених цілей необхідне вирішення наступних задач:

- розробка концепції системи і аналіз існуючих бізнес процесів;
- визначення функцій і форми представлення системи;
- проектування архітектури та розробка програмного забезпечення, бази даних;
- запропонування математичних засад формування підтримки вибору;
- проведення тестування розробленої системи на якість.

INFORMATION SUPPORT, RECOMENDATION, GRADUATION
THESIS, SPA, FIREBASE, REACT

					ДП ІС-5110.1181-с.ПЗ				
		Прізвище	Підпис	Дата					
Розроб.	Как С.Р				Інформаційна підтримка вибору наукового керівника кваліфікаційної роботи	Літ.	Лист	Листів	
Перевірив.	Тєлишева Т.О.						2		
						КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51			
Н. кон.	Тєлишева Т.О.								
Затв.	Павлов О.А.								

ABSTRACT

Structure and scope of work. Explanatory note of the diploma project consists of five sections, contains 20 drawings, 12 tables, 1 application, 10 sources.

The thesis is devoted to the creation of information support for the selection of a graduation thesis manager.

The main goals of developing a system of information support are:

- a conscious and motivated search for a thesis manager;
- organizing a comfortable working process for the student and the manager;
- improving the quality of qualification work due to joint motivated work and the interest of both parties in further development of cooperation;
- creating a comprehensive list of potential managers.

To achieve these goals, the following tasks need to be completed:

- development of the concept of the system and analysis of existing business processes;
- definition of functions and the form of system representation;
- architectural design and software development, databases;
- proposal of mathematical principles of formation choice support;
- testing of the developed system for quality.

INFORMATION SUPPORT, RECOMENDATION, GRADUATION THESIS, SPA, FIREBASE, REACT.

					ДП ІС-5110.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І	
ТЕРМІНІВ.....	5
ВСТУП.....	6
1 ЗАГАЛЬНІ ПОЛОЖЕННЯ	7
1.1 ОПИС ПРЕДМЕТНОГО СЕРЕДОВИЩА	7
1.1.1 Опис процесу діяльності	7
1.1.2 Опис функціональної моделі	9
1.2 ОГЛЯД НАЯВНИХ АНАЛОГІВ	14
1.3 ПОСТАНОВКА ЗАДАЧІ.....	16
1.3.1 Призначення розробки	16
1.3.2 Цілі та задачі розробки	16
Висновок до розділу	17
2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ	18
2.1 ВХІДНІ ДАНІ	18
2.2 ВИХІДНІ ДАНІ.....	19
2.3 ОПИС СТРУКТУРИ БАЗИ ДАНИХ	21
Висновок до розділу	23
3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ	24
3.1 ЗМІСТОВНА ПОСТАНОВКА ЗАДАЧІ	24
3.2 МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ	27
3.3 ОБГРУНТУВАННЯ МЕТОДУ РОЗВ'ЯЗАННЯ.....	28
3.4 ОПИС МЕТОДІВ РОЗВ'ЯЗАННЯ	29
Висновок до розділу	32
4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ	33
4.1 ЗАСОБИ РОЗРОБКИ	33
4.2 ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ	36
4.2.1 Загальні вимоги	36
4.3 АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	37
4.3.1 Діаграма класів	38

4.3.2	Діаграма послідовності.....	38
4.3.3	Специфікація функцій.....	39
	Висновок до розділу	41
5	ТЕХНОЛОГІЧНИЙ РОЗДІЛ	42
5.1	КЕРІВНИЦТВО КОРИСТУВАЧА	42
	Перегляд керівників	42
	Отримання рекомендованого викладача.....	43
	Додавання нового керівника адміністратором.....	45
5.2	ВИПРОБУВАННЯ ПРОГРАМНОГО ПРОДУКТУ	48
5.2.1	Мета випробувань	48
5.2.2	Загальні положення	48
5.2.3	Результати випробувань	48
	Висновок до розділу	53
	ЗАГАЛЬНІ ВИСНОВКИ.....	54
	ПЕРЕЛІК ПОСИЛАНЬ	55
	ДОДАТОК А	57

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ**

БД – база даних

SPA – Single Page Application, односторінкове веб-застосування

UML – уніфікована мова моделювання

КР – атестаційна робота для здобуття кваліфікації першого освітнього рівня бакалавр

ДП – дипломне проектування

Firebase – хмарна платформа розробки мобільних та веб-застосунків

JS – мова програмування JavaScript

React – JS-бібліотека для розробки SPA

Redux – архітектурний шаблон управління потоком даних

					ДП ІС-5110.1181-с.ПЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

При написанні кваліфікаційної роботи значну роль відіграє науковий керівник, адже він керує науковою роботою студентів, розробляє тематику дипломного проектування, рекомендує необхідну літературу, дає методичні вказівки та рекомендації.

У дійсності кафедра підтримує становище, коли студент залучений до процесу пошуку і вибору керівника дипломного проекту. Проте ця підтримка є досить обмеженою, не зважаючи на те, що вдалий вибір може привести не тільки до успішного захисту, а і до розвитку професійних навичок студента в результаті співпраці. Отже, доцільним є створення системи, яка дозволить скоротити час вибору керівника з точки зору усвідомленої взаємодії.

Дипломний проект присвячений розробці системи інформаційної підтримки, яку можна вважати рекомендаційною. Розширення інформаційного поля студентів стосовно напрямів у яких працюють керівники дозволяє студенту зробити усвідомлений вибір, що збільшує вмотивованість до підвищення якості виконання кваліфікаційних робіт, як з боку студента, так і керівника.

Практичне значення одержаних результатів.

Розроблено веб-застосування для розширення інформаційного поля студентів про напрями наукової і практичної діяльності керівників та отримання рекомендації вибору дипломного керівника.

					ДП ІС-5110.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Опис предметного середовища

При написанні кваліфікаційної роботи велику роль відіграє науковий керівник, адже він володіє методологією виконання робіт і підтримує студента у виконанні графіка.

В процесі вибору дипломного керівника можна виділити наступні категорії студентів-дипломників:

- студент з молодших курсів має певну сферу інтересів, у якій активно розвивається. Доцільним для такого студента буде керівник, досвід та знання якого, зможуть допомогти у подальшому розвитку в цій області знань;
- студент рівномірно розвивається у кількох напрямках, але не знайшов керівника. Йому потрібен керівник який буде для нього наставником, допоможе систематизувати знання та направити їх у правильне русло;
- студенти-дипломники які прагнуть отримати практичний досвід у виробничих компаніях, здобути перше робоче місце.

Наприклад, маємо студента N, який з другого семестру першого курсу почав цікавитись напрямом штучних нейронних мереж. Свою дипломну роботу бакалавра він бажає писати на тему «Побудова та дослідження мереж адаптивного резонансу». Доцільним буде приставити такого студента до керівника, який займається дослідницькою діяльністю у цій сфері та має практичний досвід застосування цих знань.

1.1.1 Опис процесу діяльності

На сьогоднішній день процес вибору дипломного керівника відбувається по різному, а прийняте рішення не завжди приводять до бажаних результатів.

					ДП ІС-5110.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

Категорії, виділені в пункті 1.1 мають свої переваги та проблеми. Продемонструємо їх у таблиці 1.1.

Таблиця 1.1 – Опис категорій студентів при виборі керівника

Переваги	Проблеми
<i>Перша категорія</i>	
Має можливість вибору дипломного керівника під свій інтерес.	Може помилитися, адже розвивався в єдиному напрямку.
<i>Друга категорія</i>	
Має високий рівень знань у базових дисциплінах і добру репутацію серед викладачів.	Не має спрямованості. Потребує керівника, який сам запропонує йому тему для кваліфікаційної роботи і виявить здібності до конкретного напрямку або області знань.
<i>Третя категорія</i>	
Має початкові практичні навички, цілеспрямований.	Виникають складності з відвідуванням очних консультацій. Можливо потребує більш гнучкого графіка, тому бажав би працювати з керівником, який практикує дистанційну роботу з студентом.

До опису переваг та недоліків можна додати, що керівники не мають актуальної інформації про студентів, тому часто будують своє враження на основі тих досягнень, які студент має з дисципліни викладача.

Вибір студентом керівника, робота з яким дозволить досягти високого рівня результату, ускладнюється ще більше у випадку пошуку варіанту серед керівників, які не викладали у студента-дипломника.

Таким чином, виходячи з аналізу цього процесу можна зробити висновок, що інформаційна підтримка процесу вибору наукового керівника є доцільною.

Покажемо як відбувається процес вибору дипломного керівника на поточний момент за допомогою діаграми IDEF0. Концептуальна схема моделі зображена на рисунку 1.1.

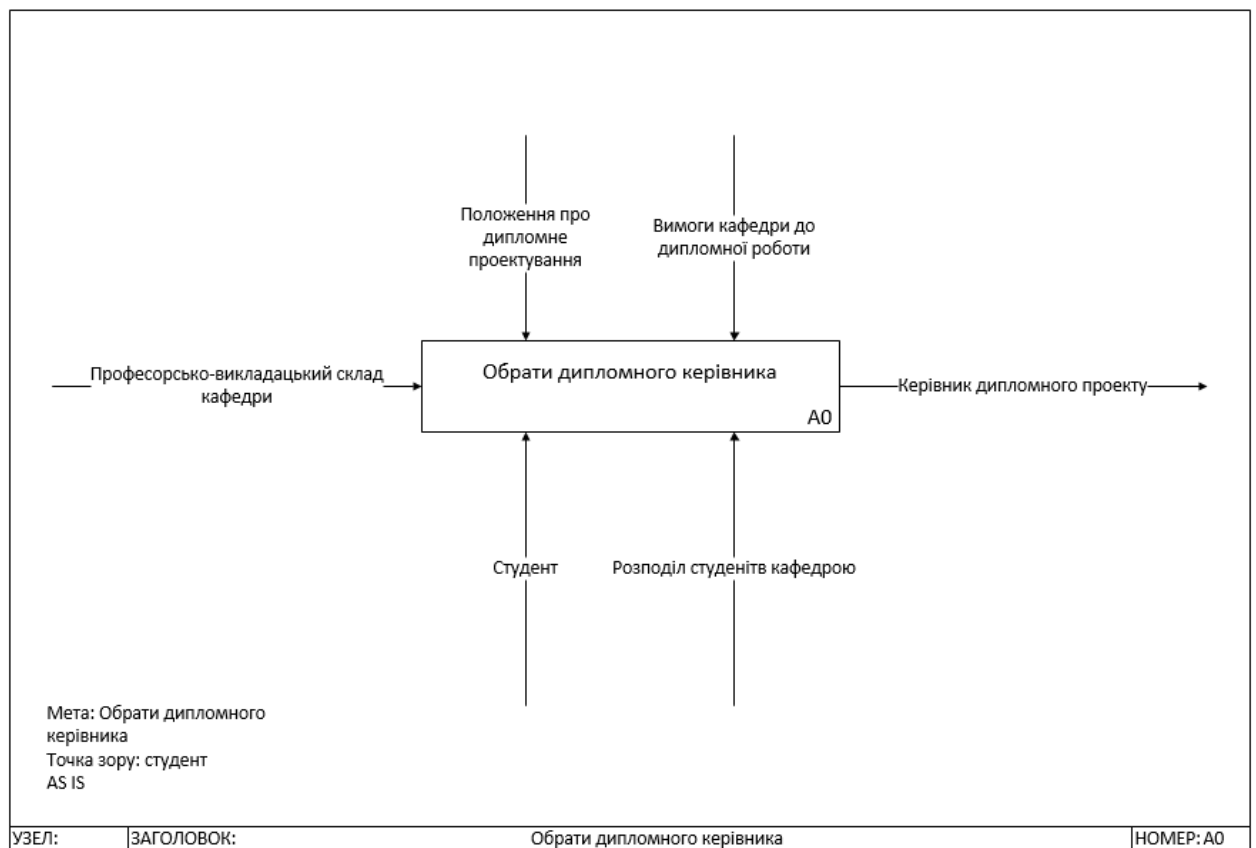


Рисунок 1.1 – Схема концептуальна функціональної моделі IDEF0

1.1.2 Опис функціональної моделі

У системі, яку можна рахувати як рекомендаційну і яка реалізована в формі веб-застосування, будуть взаємодіяти два актори: працівник кафедри і студент.

Студент – дипломник, який шукає керівника для написання бакалаврської роботи.

Працівник кафедри – співробітник кафедри, відповідальний за розподіл студентів між дипломними керівниками.

Опис функцій працівника кафедри наведено у таблиці 1.2.

Таблиця 1.2 – Опис функцій працівника кафедри

Актор	Функція	Опис функції
Працівник кафедри	1. Надання інформації про керівника	Працівник кафедри додає у систему нового керівника, вводить усю необхідну інформацію по ньому (Ім'я, Прізвище, Контактну електронну адресу) та додає фото.
	2. Визначення області зацікавленості керівника у науково-технічних сферах	Працівник кафедри обирає рівень компетенції керівника за напрямками наукової діяльності у відповідності до шкали (не зацікавлений, зацікавлений, є досвід)
	3. Задання зручного для керівника режиму робочого процесу (співпраці)	Працівник кафедри може задати зручний для керівника режим співпраці у відповідності до шкали (дистанційний, очний, посередній)

Опис функцій студента наведено у таблиці 1.3.

Таблиця 1.3 – Опис функцій студента

Актор	Функція	Опис функції
Студент	1. Перегляд профілів керівників	Студент може переглядати інформацію, надану керівниками у доступ до студента
	2. Задання зацікавленості науковими напрямками	Студент може задати зацікавленість науковим напрямом відповідно до шкали (не зацікавлений, зацікавлений, є досвід)
	3. Задання формату зустрічей	Студент може задати формат зустрічей відповідно до шкали (дистанційний, очний, посередній)
	4. Отримання рекомендації щодо керівника	Студент може отримати рекомендацію щодо керівника, переглянути сумісність інтересів та формату роботи з керівниками
	5. Формування шаблону листа для керівнику	Студент може сформувати шаблон листа для комунікації з керівником

На рисунку 1.2 зображено структурну схему варіантів використання. Акторами на діаграмі виступають студент і працівник кафедри. Студент може переглядати профілі керівників, отримувати рекомендацію щодо вибору керівника, сформувати e-mail листа. Працівник кафедри може створювати профіль керівника відповідно до даних, які надає керівник.

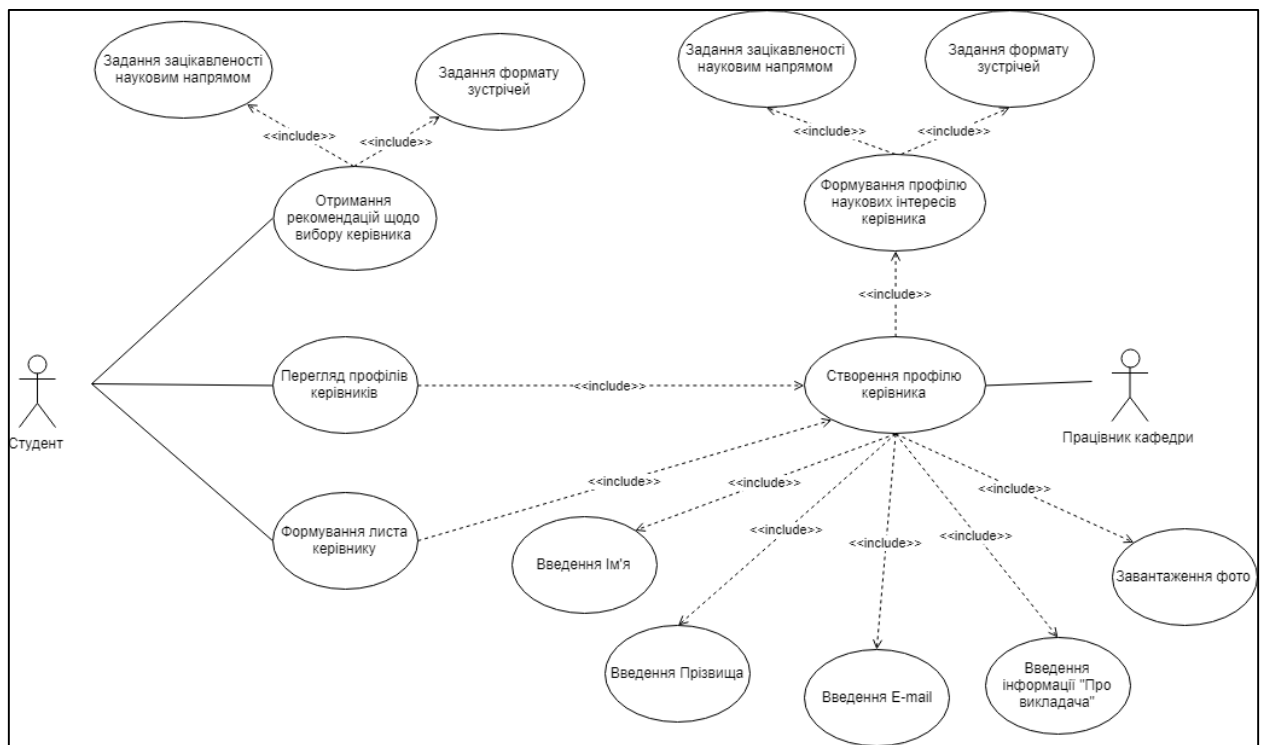


Рисунок 1.2 – Схема структурна варіантів використання

Покажемо як може відбуватись процес пошуку дипломного керівника з використанням веб-застосування інформаційної підтримки за допомогою наступних діаграм діяльності з доріжками. Учасниками процесів є студент та працівник кафедри, відповідальний за розподіл студентів між керівниками.

Основними етапами діяльності студента є вхід до веб-сайту, перегляд профілів керівників, визначення зручного для студента формату роботи з керівником, отримання рекомендації щодо керівника. Діаграма діяльності керівника наведена на рисунку 1.3.

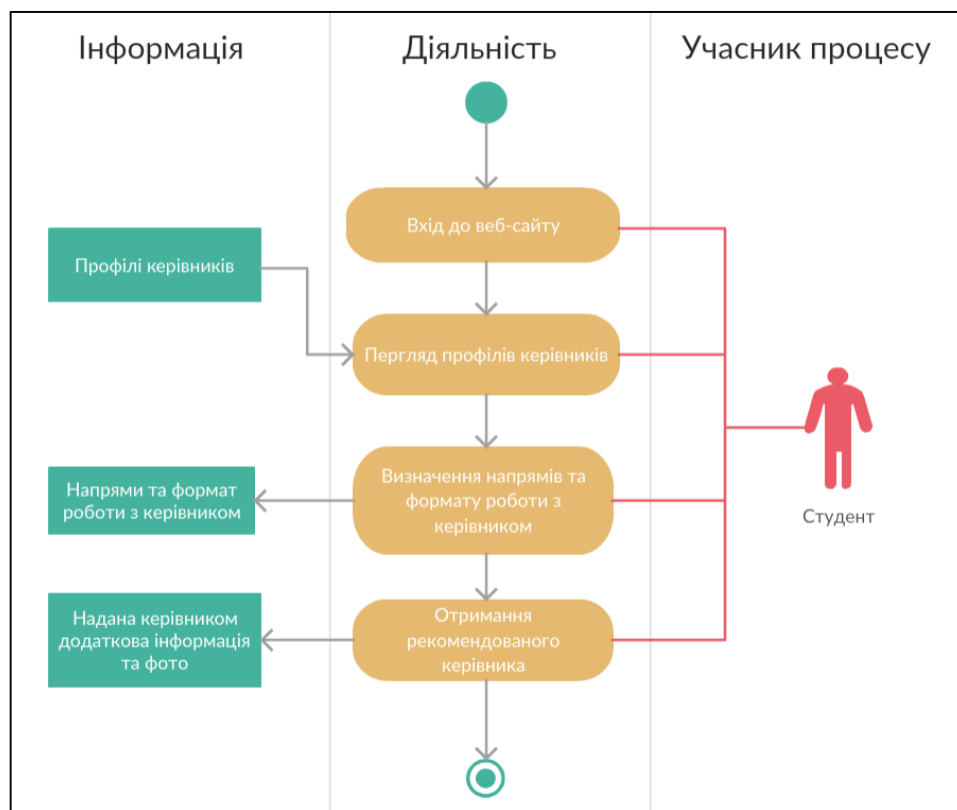


Рисунок 1.3 – Діаграма діяльності студента

Основні етапи діяльності працівника кафедри описані на діаграмі діяльності з доріжками на рисунку 1.4.

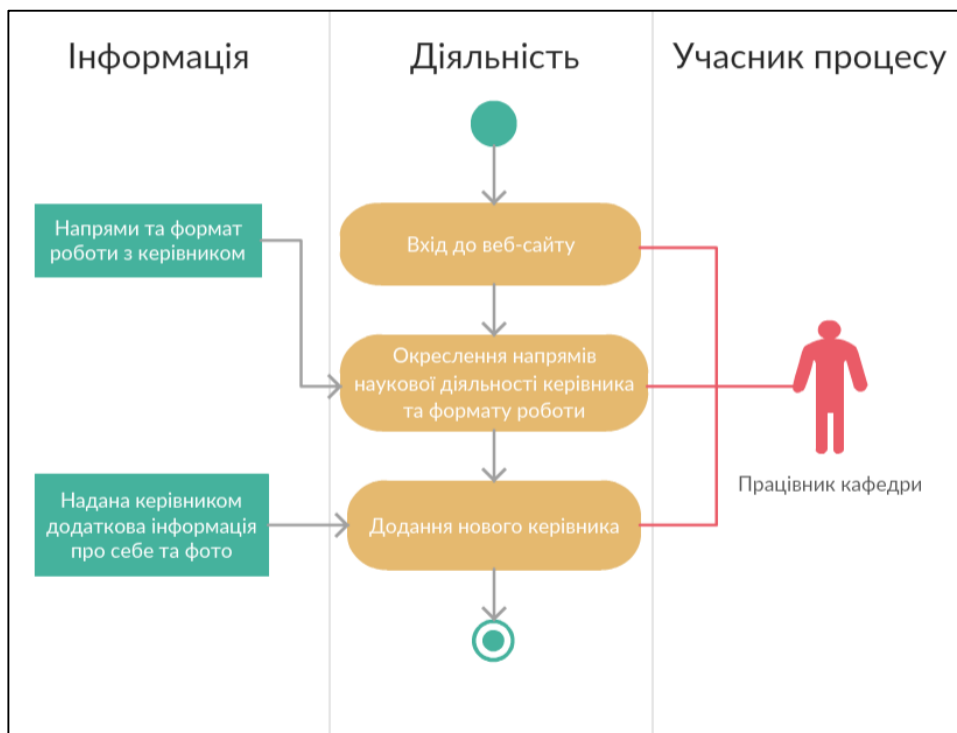


Рисунок 1.4 – Діаграма діяльності керівника

1.2 Огляд наявних аналогів

У процесі пошуку аналогів не було виявлено інформаційних систем, які дозволяли б у будь-якій мірі допомогти у пошуку керівників та отримання рекомендацій щодо вибору. В мережі Інтернет можна знайти багато статей та порад стосовно правильного вибору дипломного керівника, але вони є суб'єктивними поглядами авторів, а поради мають загальний характер.

Подібний функціонал властивий системам, що спеціалізуються на підборі репетиторів: «Ваш репетитор», «Buki».

Таблиця 1.4 – Опис подібних сервісів

Назва	Опис
Ваш репетитор https://kiev.repetitors.info/	Ресурс надає можливість переглядати анкети викладачів, користуватись безкоштовним онлайн підбором репетитора. Крім цього, є можливість зареєструватись у базі викладачів та розмістити свою анкету.
Buki https://buki.com.ua/	Сервіс допомагає учням пришвидшити знаходження репетиторів та вчителів, шляхом самостійного пошуку серед анкет або за допомогою адміністратора, який пропонує найбільш підходящих кандидатів. До переваг сервісу відносяться: велика база репетиторів, рейтингова система за відгуками учнів, охоплення понад 100 предметів та більш ніж 200 міст України.

Інтерфейс сервісу «Buki», який має схожу мету, але серед репетиторів, наведено зображено на рисунку 1.5.

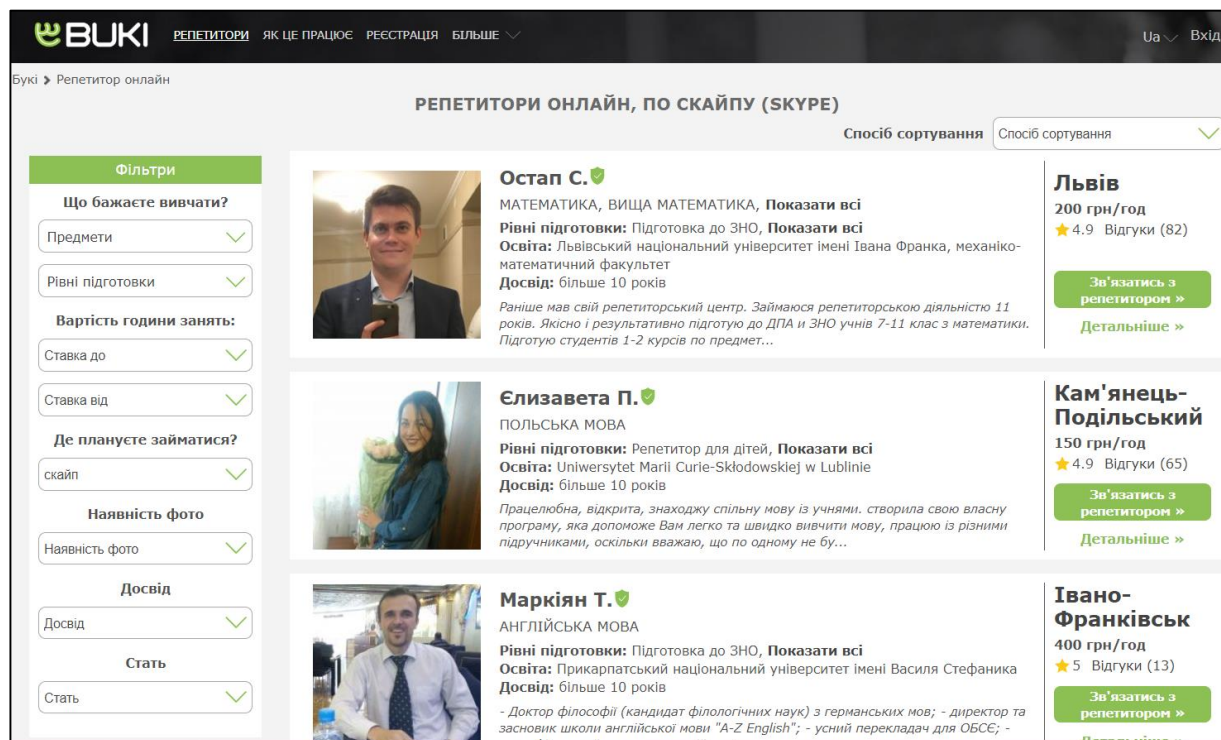


Рисунок 1.5 – Інтерфейс сервісу «Buki»

Інтерфейс сервісу «Ваш репетитор» наведено зображено на рисунку 1.6.

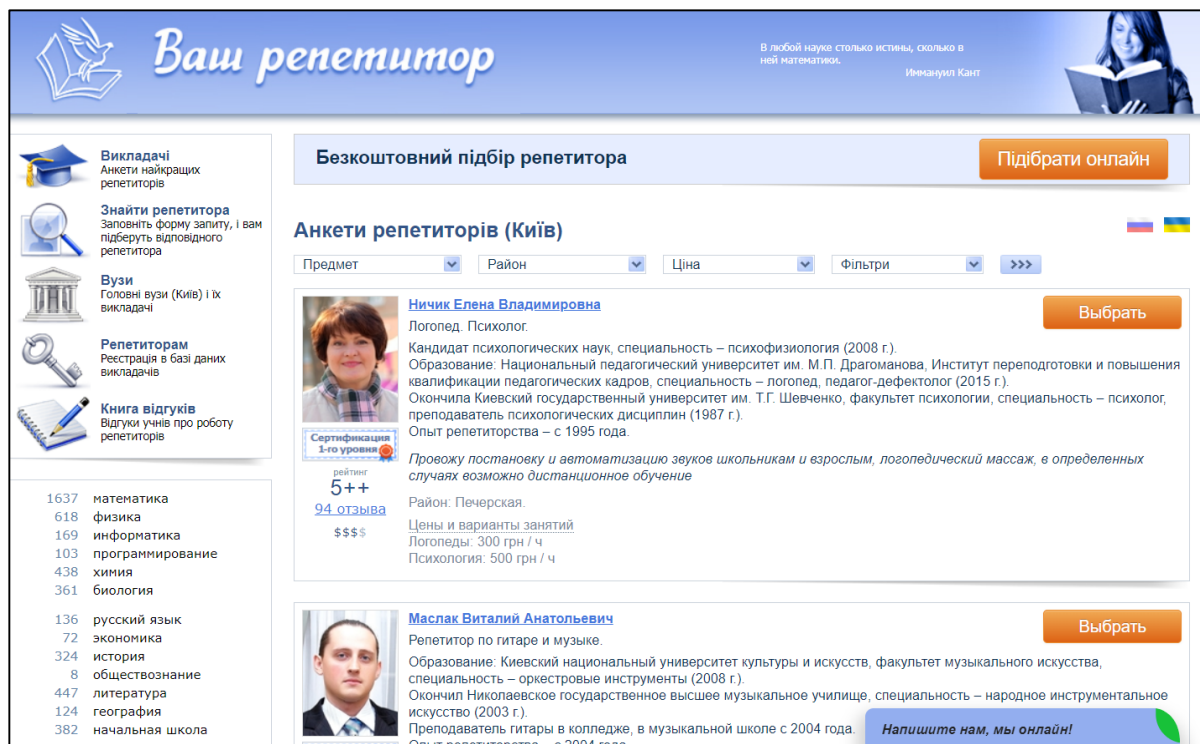


Рисунок 1.6 – Інтерфейс сервісу «Ваш репетитор»

Змн.	Арк.	№ докум.	Підпис	Дата

1.3 Постановка задачі

1.3.1 Призначення розробки

Система інформаційної підтримки призначена для полегшення процесу вибору студентом дипломного керівника. Веб застосування допомагає студенту знайти компетентного викладача.

1.3.2 Цілі та задачі розробки

Основна ціль системи інформаційної підтримки вибору наукового керівника – дозволити студенту зробити усвідомлений вибір керівника кваліфікаційної роботи, що збільшить вмотивованість до підвищення якості виконання кваліфікаційної роботи, як з боку студента, так і керівника.

Для досягнення мети необхідне вирішення наступних задач:

- розробка концепції системи і аналіз існуючих бізнес процесів;
- визначення функцій і форми представлення системи;
- проектування архітектури та розробка програмного забезпечення, бази даних;
- запропонування математичних засад формування підтримки вибору;
- проведення тестування розробленої системи на якість.

При вирішенні зазначених задач будуть досягнуті наступні цілі:

- усвідомлений та мотивований пошук керівника;
- впорядкування комфортного робочого процесу для студента і керівника;
- підвищення якості кваліфікаційних робіт внаслідок сумісної мотивованої роботи та зацікавленості обох сторін у подальшому розвитку співпраці;

- створення єдиного розгорнутого списку потенціальних керівників.

Висновок до розділу

У розділі описано предметне середовище, у якому виділено категорії студентів у контексті вибору дипломного керівника. Описано процес діяльності, у якому порівнюються відповідні категорії. Також побудовано концептуальну схему IDEF0 процесу вибору дипломного керівника на момент відсутності веб застосунку.

У підрозділі опису функціональної моделі визначено акторів веб застосування, побудовано структурну схему варіантів використання. Проведено огляд систем з подібним функціоналом та виявлено, що аналогів подібному застосуванню немає. Виходячи з цього, сформовано цілі та задачі розробки такого застосування.

Надана ідея, при вдалій реалізації, може слугувати корисним засобом для студентів, які займаються пошуком дипломного керівника, адже наразі процес пошуку відбувається по різному і не завжди приводить до бажаних результатів.

2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Вхідні дані

Надаються керівником для працівника кафедри

Контактна інформація:

- Ім'я
- Прізвище
- Електронна адреса

Додаткова інформація про керівника, сформована ним самим і надана в доступ студенту, а також фото. На рисунку 2.1 наведено приклад введення цих вхідних даних.

Рисунок 2.1 – Приклад введення вхідних даних керівника

Надаються і студентом і керівником:

Зацікавленість у наступних науково-технічних сферах (не зацікавлений, зацікавлений, є досвід):

- Алгоритми (Algorithms);
- Математичне моделювання (Scientific modelling);
- Веб технології (Web technologies);
- Мобільні технології (Mobile technologies);
- Штучний інтелект (Artificial intelligence);

Побажання до робочого процесу:

- Формат зустрічей (more distance – equal – more face to face);

На рисунку 2.2 наведено приклад введення цих вхідних даних.

Interests:

Algorithms
Not interested ▼

Scientific Modeling
Not interested ▼

Web Technologies
Not interested ▼

Mobile Technologies
Not interested ▼

Artificial Intelligence
Not interested ▼

Working format:

☒ More distance

☐ Equal

☐ More Face to Face

UPDATE

Рисунок 2.2 – Приклад введення вхідних даних студента

2.2 Вихідні дані

Вихідними даними є сформований список профілів керівників, який дозволяє студенту здійснити попередній аналіз майбутньої роботи з потенційними керівниками, а також рекомендовані. Приклад фрагменту вихідних даних наведено на рисунку 2.3.

**Валентина Плаксі**

Доцент, кандидат наук. Веду дослідницьку діяльність у сфері кластерного аналізу.

E-mail: vplox@gmail.com

Interests:

- algorithms
- ai

Choosed: 0 times.

**Іван Петров**

Професор. Науковий напрям - алгоритмізація та програмування. Практикую дистанційну роботу зі студентами.

E-mail: ipetrov@yahoo.com

Interests:

- algorithms
- web

Choosed: 0 times.

**Сергій Іванов**

Старший викладач. Маю досвід практичної роботи з системами штучного інтелекту.

E-mail: sivanov@gmail.com

Interests:

- algorithms
- ai

Choosed: times.

**Тарас Клименок**

Кандидат наук. Маю практичний досвід та глибокі знання у розробці мобільних архітектур.

E-mail: tklim@meta.ua

Interests:

- web
- mobile

Choosed: 0 times.

**Людмила Ткач**

Старший викладач. Сфера діяльності - алгоритми та моделювання.

E-mail: ltkach@gmail.com

Interests:

- algorithms
- modeling

Choosed: 6 times.

Рисунок 2.3 – Приклад вихідних даних

Змн.	Арк.	№ докум.	Підпис	Дата

2.3 Опис структури бази даних

Оскільки функціонал веб-застосунку у майбутньому може бути розширений для студентів-дипломників інших кафедр та факультетів, використання баз даних з можливістю легкого масштабування є доцільним. У даній роботі розглядаються керівники та студенти кафедри, навчальний план якої передбачає вивчення дисциплін науково-технічних сфер, вказаних у підрозділі вхідних даних. У якості бази даних обрано хмарну NoSQL базу даних Firebase Realtime Database[1], яка надає можливість отримання актуальних даних БД у режимі реального часу.

NoSQL[2] сховища оперують з даними як з цілісними об'єктами на відміну від реляційних моделей, котрі зберігають сутності у різні фізичні таблиці з метою нормалізації. Зважаючи на те, що звичних для реляційної бази даних таблиць немає, опишемо об'єкт основної колекції «Керівники», яка зберігає дані про керівників у таблиці 2.1.

Таблиця 2.1 – Структура об'єкта «Керівник»

Назва	Опис	Тип даних
Id	Унікальний ідентифікатор керівника	String
Name	Ім'я керівника	String
Surname	Прізвище керівника	String
Email	Електронна адреса керівника	String
ImageURL	Посилання на фотографію керівника, яка міститься у сховищі Firebase Storage	String
Interests	Масив зацікавленостей керівника у науково-технічних сферах відповідно до вхідних даних	List<Interest>
Working Format	Зручний для керівника формат зустрічей	Number

База даних зберігає профілі викладачів у форматі JSON у деревовидній структурі. Фрагмент заповненої керівниками бази даних наведено на рисунку 2.2.

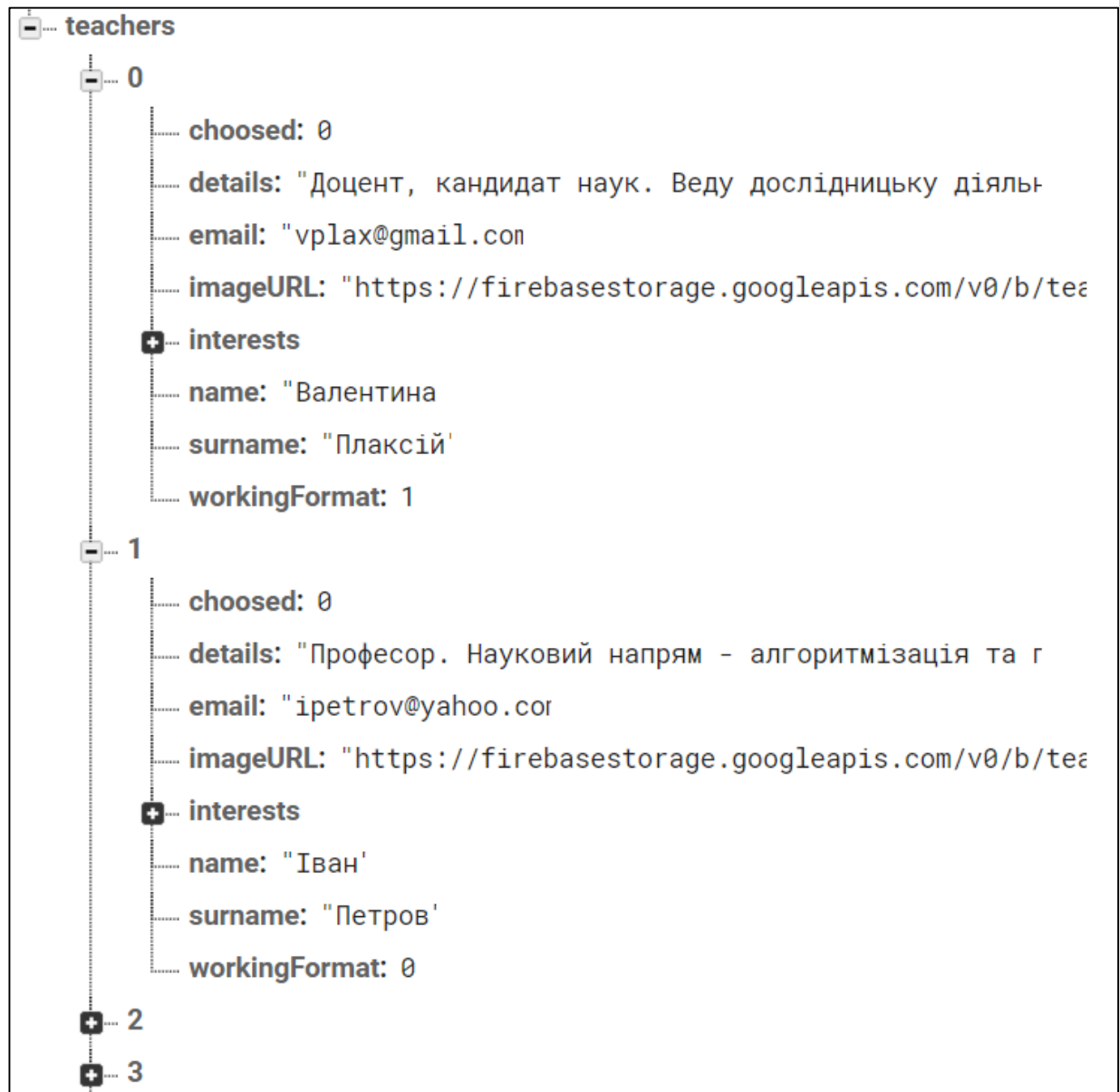


Рисунок 2.4 – Фрагмент заповненої керівниками БД

Висновок до розділу

У цьому розділі розглянуто вхідні та вихідні дані, які потрібні для збереження даних про керівника та формування його профілю, наведено приклади заповнення. Так як необхідним для веб застосування є надання доступу до профілів керівників багатьом студентам, доцільним є збереження даних у централізовану БД.

У якості такої бази було обрано хмарну базу даних Firebase Realtime Database яка надає широкі можливості для розробника, серед яких синхронізація між усіма клієнтами БД в режимі реального часу.

Крім цього важливим для нашої бази є горизонтальне масштабування у випадку додання нових областей знань для студента та керівника, тому доцільним буде використовувати NoSQL підхід до організації та проектування бази даних.

Коректність введених даних контролюється шляхом введення спеціальних правил та обмежень на властивості об'єктів колекції. Сервер бази даних слідкує за внесенням даних до бази та при некоректному введенні видає помилку, яка перехоплюється веб-застосунком.

3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Змістовна постановка задачі

Захист атестаційної роботи є необхідною умовою для здобуття документа державного зразка про отримання вищої освіти. До факторів, які впливають на успішний захист можна віднести вибір теми та керівника атестаційної роботи. У багатьох європейських університетах виконання атестаційної роботи може консультувати декілька викладачів у відповідності зі змістом і темою роботи. Він перевіряє своєчасне виконання наукових та навчальних завдань, працює у тісному контакті з підопічними, тому студенти із задоволенням користуються його досвідом та знаннями.

Обов'язки та взаємовідносини між керівником і студентом регулюють методичні та юридичні документи. Так, наприклад методичні вказівки до виконання дипломних проектів (робіт) першого (бакалаврського) рівня вищої освіти «Бакалавр» [3] передбачають, що для керівництва студентами, які мають підготувати атестаційні роботи, призначаються науково-педагогічні працівники випускової кафедри, а також провідні співробітники наукових підрозділів кафедри або провідні спеціалісти у відповідній галузі з підприємств, науково-дослідних інститутів, міністерств, відомств тощо. За викладачами, які здійснюють керівництво студентами вперше, можуть за рішенням кафедри закріплюватися консультанти – досвідчені науково-педагогічні працівники кафедри із зазначенням часу, який вони витрачають у розділі «Методична робота» індивідуального плану (але не більше 20 годин).

За рішенням кафедри або на прохання керівника можуть призначатися консультанти:

- зі специфічних виробничих, технічних, наукових питань;
- питань, які відносяться до компетенції кафедр фундаментальних чи професійно-орієнтованих дисциплін;

- техніко-економічного обґрунтування прийнятих рішень та розрахунків економічного ефекту;
- питань екології, безпеки життєдіяльності та охорони праці тощо.

Список консультантів за поданням випускової кафедри затверджується розпорядженням директора інституту/декана факультету.

У реальному житті кафедра підтримує становище, коли студент залучений до процесу вибору керівника проекту. Але підтримка вибору досить обмежена, не зважаючи на те, що правильний вибір може привести не лише до вдалого захисту, а і до подальшої співпраці та розвитку професійних навичок студента. При спільній роботі студента і керівника важливим є збіг зацікавленості напрямом, за яким відбувається дослідження або розробка, спосіб комунікації та таке інше. Так, встановлення правильних ділових відносин між керівником і студентом спричиняє підвищення якості атестаційної роботи в результаті покращення взаєморозуміння.

Для формалізації та автоматизації процесів підтримки вибору керівника в дипломному проекті було виділено напрями, за якими, в основному, можна поділити загальний об'єм знань і навичок, які повинен підтвердити дипломник в атестаційній роботі.

Algorithms (AL) – розробка алгоритмів. Напрямок, який полягає у створенні покрокових процедур або наборів інструкцій, послідовне виконання яких призводить до досягнення поставленої мети за визначений час.

Scientific Modeling (SM) – спосіб дослідження [4] будь-яких явищ, процесів або об'єктів шляхом побудови та аналізу їх моделей. Серед видів моделювання основними є комп'ютерне, математичне, імітаційне та статистичне.

Web-technologies (WT) – загальна назва для спрямування, яке націлене на розробку веб-сайтів, веб-застосовувань. Також до веб – технологій відносимо

розробку мережевих протоколів та дослідження галузей, які стосуються побудови обчислювальних мереж, наприклад Internet of Things та інші.

Mobile-technologies (MT) – технології, пов’язані з електронним обладнанням, таким як мобільні телефони або невеликі комп’ютери, які можна використовувати у різних місцях.

Artificial-Intelligence(AI) – напрям, який ототожнює задачі відтворення тих чи інших когнітивних міркувань та дій за допомогою розрахункових систем. Включає в себе напрями машинного навчання, створення штучних нейронних мереж та інші.

Результатами проведення опитування 100 студентів третього курсу, які ще не обрали дипломного керівника, щодо зацікавленості роботи над дипломним проектом у одному з зазначених вище напрямів наведена на рисунку 3.1.

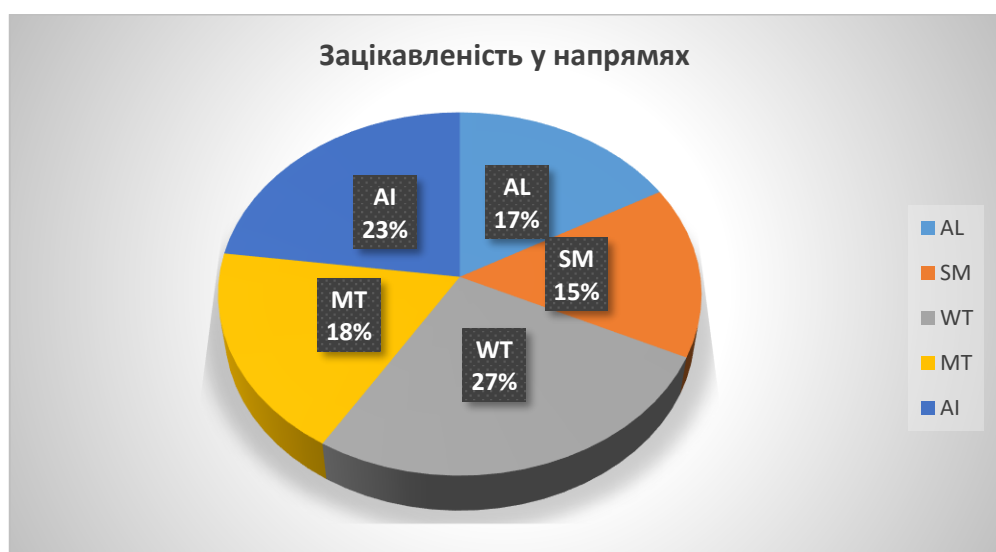


Рисунок 3.1 – Зацікавленість студентів у напрямках

Можна зробити висновок про доцільність введення цих напрямів, адже жоден з них не залишився без зацікавленого студента.

Серед критеріїв, які є важливими у ділових стосунках між студентом і керівником, найбільш важливими є:

- збіг зацікавленості студента і керівника у напрямках дипломної роботи, які було виділено у процесі дослідження (AL, SM, WT, MT, AI);
- взаємна зацікавленість студента і викладача у форматі консультацій (WF- working format).

Таким чином, задача зводиться до підбору найбільш доречного керівника за описаними вище критеріями.

3.2 Математична постановка задачі

Призначенням цієї задачі є знаходження для студента найбільш доречного, з точки зору майбутньої співпраці керівника. Вхідні дані до даної задачі надані в таблиці 3.2

В таблиці вхідних даних в якості «Об'єкт» наданий перелік викладачів, які за наказом можуть стати керівниками дипломних проектів, їх основні сформульовані напрямки, а також формат консультацій (WF- working format). Об'єкт студента у цій таблиці має аналогічні до керівника ознаки.

Таблиця 3.2 – Вхідні дані

Об'єкт	AL	SM	WT	MT	AI	WF
Керівник 1	$p_{1,1}$	$p_{1,2}$	$p_{1,3}$	$p_{1,4}$	$p_{1,5}$	$p_{1,6}$
Керівник 2	$p_{2,1}$	$p_{2,2}$	$p_{2,3}$	$p_{2,4}$	$p_{2,5}$	$p_{2,6}$
Керівник 3	$p_{3,1}$	$p_{3,2}$	$p_{3,3}$	$p_{3,4}$	$p_{3,5}$	$p_{3,6}$
...
Керівник N	$p_{n,1}$	$p_{n,2}$	$p_{n,3}$	$p_{n,4}$	$p_{n,5}$	$p_{n,6}$
Студент	q_1	q_2	q_3	q_4	q_5	q_6

Виходячи з наданої таблиці можемо виділити:

- Набір ознак студента $q_j, j = \overline{1,6}$;
- Набір ознак і-го керівника $p_{i,j}, j = \overline{1,6}$.

Зацікавленість і-го керівника у j-му напрямі $p_{i,j} (j = \overline{1,5})$ може набувати значень:

$$p_{i,j} = \begin{cases} 0, \text{ якщо } i - \text{й керівник не цікавиться } j - \text{м напрямом,} \\ 1, \text{ якщо } i - \text{й керівник цікавиться } j - \text{м напрямом,} \\ 2, \text{ якщо } i - \text{й керівник має досвід у } j - \text{му напрямі.} \end{cases} \quad (3.1)$$

Зацікавленість студента у j-му напрямі $q_j (j = \overline{1,5})$ може набувати значень:

$$q_j = \begin{cases} 0, \text{ якщо студент не цікавиться } j - \text{м напрямом,} \\ 1, \text{ якщо студент цікавиться } j - \text{м напрямом,} \\ 2, \text{ якщо студент має досвід у } j - \text{му напрямі.} \end{cases} \quad (3.2)$$

Зручний для і-го керівника формат консультацій $p_{i,6}$ може набувати значень:

$$p_{j,6} = \begin{cases} 0, \text{ якщо керівник обирає } More Distance, \\ 1, \text{ якщо керівник обирає } Equal, \\ 2, \text{ якщо керівник обирає } More Face to Face. \end{cases} \quad (3.3)$$

Зручний для студента формат консультацій q_6 може набувати значень:

$$q_6 = \begin{cases} 0, \text{ якщо студент обирає } More Distance, \\ 1, \text{ якщо студент обирає } Equal, \\ 2, \text{ якщо студент обирає } More Face to Face. \end{cases} \quad (3.4)$$

Необхідно визначити викладача, який найбільше задовольняє побажанням студента в якості керівника дипломного проекту.

3.3 Обґрунтування методу розв'язання

Кластерний аналіз – термін, вперше введений Р.Тріоном[5], є задачею багатовимірного статистичного збору та впорядкування даних, які містять інформацію про вибірку об'єктів. Задача кластеризації відноситься до

статистичної обробки, а також до широкого класу задач навчання без вчителя.

В дійсності ж кластерний аналіз включає в себе набір різних алгоритмів класифікації, серед яких:

- ієрархічна класифікація;
- деревовидна класифікація;
- метод К середніх та багато інших.

Фактично, кластерний аналіз є не стільки загальним статистичним методом, скільки набором різних алгоритмів розподілення об'єктів по кластерам. На відміну від багатьох інших статистичних процедур (наприклад дисперсійного аналізу), методи кластерного аналізу використовуються коли будь-яких апріорних гіпотез стосовно класів немає, тобто дослідження знаходиться на стадії опису.

Отже, кластерний аналіз визначає найбільш можливо значиме рішення задачі. З цього випливає, що перевірка статистичної значущості є незастосовною.

Більшість алгоритмів класифікації починаються зі знаходження відстані між об'єктами. Саме цю задачу і необхідно розв'язати для визначення відстані між студентом і керівниками. Таким чином першочерговою задачею є знаходження відстані між об'єктами. Отже задача зводиться до виміру відстаней між ознаками двох об'єктів: студента і керівника. Об'єкт представляється у вигляді точки у просторі з шести вимірів.

3.4 Опис методів розв'язання

У нашому випадку множиною на якій будуть проводитись виміри виступає множина керівників та студент. Для того щоб відповісти на питання наскільки студент і конкретний керівник відрізняються можна розрахувати відстань між ними і у відношенні до максимальної відстані представити рекомендацію наскільки студент і керівник підходять один до одного.

					ДП ІС-5110.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

Для цього необхідно визначити метрику – числову функцію $d(p, q)$, яка обраховуватиме відстань від студента q до керівника p , які є фактично точками багатовимірного простору.

Метрика Евкліда – фактично є однією з найбільш загальних метрик [5], адже визначає геометричну відстань у багатовимірному просторі за теоремою Піфагора. Відстань Евкліда є найбільш зрозумілою та інтерпретованою мірою відстані або зближення об'єктів, представлених векторами ознак у багатовимірних просторах.

$$d(p, q)_E = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (3.5)$$

Відстань міських кварталів (Манхеттенівська відстань) – метрика, яка визначає суму довжин проекцій відрізка між точками на осі координат.

$$d(p, q)_M = \sum_{i=1}^n |p_i - q_i| \quad (3.6)$$

Для розрахунку міри подібності можна також скористатись узагальненим коефіцієнтом Жаккара, який є узагальненою версією бінарної міри подібності.

$$d(p, q)_J = 1 - \frac{\sum_i^n \min(p_i, q_i)}{\sum_i^n \max(p_i, q_i)} \quad (3.7)$$

Отже, ідея алгоритму розв'язання задачі полягає у послідовному виконанні наступних кроків.

КРОК 1. Визначити зацікавленість студента і керівників науковими напрямками, описаними вище, та сформувати рядки таблиці p та q відповідно.

КРОК 2. Для кожної пари q та p_i розрахувати значення відстані за формулою (3.1) та запам'ятати значення проміжного розрахунку

$$d(p_i, q)_{prom,E} = 100 - (d(p_i, q)_E / \max(d(p, q)_E) * 100 \quad (3.8)$$


,де $\max(d(p, q))$ – максимально можлива відстань між q та p_i за обраною метрикою.

КРОК 3. Повторити дії виконані на кроці 2 для розрахунку відстані за формулою (3.6) та запам'ятати значення проміжного розрахунку $d(p_i, q)_{prom, M}$, який розраховується аналогічно до (3.8)

КРОК 4. Для кожної пари q та p_i розрахувати міру подібності за формулою (3.7) та запам'ятати значення проміжного розрахунку $d(p_i, q)_{prom, J}$.

КРОК 5. Відсортувати керівників за спаданням середнього значення отриманих на кроках 1-4 розрахунків. За розробленою концепцією алгоритму перший керівник буде рекомендованим.

На рисунку 3.2 покажемо приклад результату роботи алгоритму у реалізованому програмному продукті.



Тетяна Попова
 Старший викладач. Науковий напрям - штучний інтелект. Працював більш ніж у 20 проектах у міжнародних компаніях.
 E-mail: tpopova@gmail.com
 Interests:
 o ai
 Chooosed: 9 times.

Similarities:
 Eucludian : 80.849 %
 Manhattan : 88.889 %
 Jaccard : 50.000 %



Оксана Степанець
 Доцент. Маю досвід практичної роботи з хмарними сховищами.
 E-mail: oxxy@gmail.com
 Interests:
 o web
 o mobile
 Chooosed: 0 times.

Similarities:
 Eucludian : 69.720 %
 Manhattan : 83.333 %
 Jaccard : 40.000 %



Сергій Іванов
 Старший викладач. Маю досвід практичної роботи з системами штучного інтелекту.
 E-mail: sivanov@gmail.com
 Interests:
 o algorithms
 o ai
 Chooosed: times.

Similarities:
 Eucludian : 64.172 %
 Manhattan : 72.222 %
 Jaccard : 33.333 %

Рисунок 3.2 – Результат роботи алгоритму

Висновок до розділу

В даному розділі описано змістовну постановку задачі, детально розглянуто можливі методи розв'язання. Сформульовано математичну постановку задачі та надано ідею алгоритму знаходження рекомендованого керівника.

Треба зауважити, що вибір керівника може бути ще додатково підтриманий інформацією про керівників, яку викладач складає сам і надає в доступ для студента, який може одержати більш повне уявлення про керівника з опису. Можливо, це особливо важливо, коли викладач і студент не зустрічалися в процесі навчання.

4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

4.1 Засоби розробки

Для реалізації веб-застосування було використано мову програмування JavaScript[7], архітектурний шаблон React/Redux для створення SPA з використанням хмарного сервера Firebase для обслуговування бази даних Firebase Realtime Database та додаткового сховища Firebase Storage. Середовищем розробки обрано VS Code. Для збірки застосунку призначений модуль для збірки Webpack та JavaScript компілятор Babel. Для відстеження версій програмного продукту використовується git.

React JS – JavaScript бібліотека [8] з відкритим вихідним кодом, основним призначенням якої є побудова динамічних користувацьких веб інтерфейсів. React значно спростив процедуру їх створення завдяки розбиттю кожної веб сторінки на невеликі фрагменти, які називаються компонентами.

Не зважаючи на те, що React є бібліотекою, що відповідає за репрезентацію, на відміну від Angular або Vue, його основні парадигми фактично є стандартом для створення сучасних веб-застосунків і його використання не лише у якості View є цілком можливим [10]. React має наступні особливості:

- компонентний підхід. Компонент представляється як функція відображення інтерфейсу користувача в залежності від його стану. Інтерфейс користувача повинен мати якомога більше компонентів, щоб максимізувати повторне використання.
- JSX як синтаксичне розширення JavaScript. Рекомендується використовувати разом з React для опису того як компоненти повинні виглядати.

- односторонній потік даних. React використовує односторонній потік даних, тобто лише зміни даних призводять до змін складових інтерфейсу.

Основними перевагами React у порівнянні з іншими бібліотеками є:

- Virtual DOM – JavaScript об’єкт, який є легкою копією реального DOM (Document Object Model). React використовує його у програмному ядрі, використовуючи різницю між останнім оголошеним рендером і оновленою візуалізацією у віртуальній моделі інтерфейсу, застосовуючи до нього реалізацію реального інтерфейсу. Використання віртуальної копії підвищує продуктивність застосунків, оскільки робота з ним значно швидше, ніж зі звичайним DOM браузера.
- використання як на клієнтській, так і на серверній стороні. React є не лише бібліотекою створення інтерфейсу користувача, його можна вважати фреймворком / екосистемою. Станом даних веб застосування можна керувати, використовуючи такий менеджер станів, як Redux. Це абстракція від дерева компонентів, яка спрощує налагодження, щоб не мати прихованого стану в декількох місцях.

Redux - інструмент керування як станом [8] даних, так і станом інтерфейсу у веб-додатках. Він підходить для односторінкових (Single Page Application) додатків, в яких управління даними може з часом стати складним. Redux не пов'язаний з якимось певним фреймворком, і хоча розроблявся для React, його можна використати з Angular або jQuery.

Flux - це архітектурний шаблон написання програм, який Facebook використовує для створення веб-додатків. Він доповнює компоненти React, використовуючи односпрямований потік даних.

Redux частково переймає концепції Flux. Наприклад, так само як і Flux, Redux визначає концентрацію логіки оновлення даних в певному шарі програми (“stores” у Flux, “reducers” в Redux). Для того, щоб змінити дані, необхідно описати кожну мутацію як простий об’єкт (action), замість надання дозволу коду напряду змінювати інформацію.

Redux допомагає у створенні послідовних застосунків[9], які виконуються в різних середовищах і легкі у тестуванні. Крім цього, централізація логіки програми забезпечує потужні можливості, такі як скасування та повторне виконання, збережуваність стану та багато іншого.

Redux, як контейнер стану можна описати у трьох основних підходах:

- єдине джерело істини. Стан всього додатку зберігається у єдиному дереві об’єктів в межах одного об’єкту store. Це дозволяє створювати контрольовані застосунки, інспектування та тестування яких значно полегшується;
- стан програми доступний лише для читання. Єдиним способом змінити стан у застосуванні є створення дії (action), об’єкту, який описує, яким чином повинні змінитись дані. Це гарантує, що ані зміни, спричинені діями користувача, ані, так звані callbacks (зворотні виклики) мережі не змінюють дані та стан. Натомість, створюються наміри трансформації, які виконуються у відповідному порядку, таким чином позбавляючи застосування від блокуючих умов змагання за зміну даних;
- зміни через використання чистих функцій. Щоб вказати, яким чином дерево станів трансформується діями, описуються чисті функції (reducers). Чистими вважаються функції, які, отримують попередній стан застосування та дію, а повертають нові змінені дані.

Firebase – платформа, яка надає розробнику інструменти для роботи з базою даних Firebase Realtime Database з підтримкою змін у реальному часі та сервер як службу. Служба надає розробникам програм API, який дозволяє синхронізувати дані програми між клієнтами і зберігати їх в хмарі Firebase.

Firebase Real Time Database слугує рішенням, котре застосовує NoSQL підхід до зберігання та видобування даних. NoSQL забезпечує простоту дизайну схеми бази даних, спрощує масштабування, дозволяє забезпечити більш тонкий контроль над доступністю.

4.2 Вимоги до технічного забезпечення

4.2.1 Загальні вимоги

Розроблений програмний продукт є веб-застосуванням, використання якого не потребує спеціальних знань або кваліфікацій.

Для забезпечення коректної та стабільної роботи необхідне забезпечення виконання усіх наступних вимог:

1. Пристрій:

- будь-який сучасний пристрій з можливістю доступу до мережі інтернет

2. Операційні системи:

- Windows
- MacOS
- Linux

3. Веб-браузер:

- Google Chrome версії не нижче 74.0.3729.169
- Safari версії не нижче 12.1.1
- Mozilla Firefox версії не нижче 67.0

4. Доступ до мережі Інтернет

4.3 Архітектура програмного забезпечення

Архітектура веб-застосування складається з двох рівнів:

- React/Redux SPA застосування для динамічного відображення веб-сторінок та організації взаємодії з користувачем шляхом використання AJAX запитів;
- хмарний сервер Firebase для роботи з базою даних Firebase Realtime Database та Firebase Storage.

Така архітектура має наступні переваги:

- еластичність, адже провайдер послуг автоматично надає більше або менше ресурсів, в залежності від потреб розробки;
- абстракція від нюансів операційної системи, оновлень, параметрів мережі та іншого. Це зроблено для зосередження на розробці, а не на адмініструванні серверів;

Архітектура веб-застосування SPA (Single-page application) базується на понятті serverless з використанням хмарних технологій Firebase. SPA застосування дозволяють імітувати роботу десктоп додатків при цьому працюючи повністю у браузері. Основними перевагами SPA [11] є висока швидкість взаємодії, так як SPA не оновлює всю сторінку, а тільки потрібну частину, що істотно підвищує швидкість роботи.

SPA влаштовані таким чином, що при переході на нову сторінку, оновлюється тільки частина контенту. Таким чином, немає необхідності повторно завантажувати одні й ті ж елементи. Це дуже зручно для розробників і користувачів. Легковажні за часом очікування операції відбуваються на стороні SPA, а виконання важких операцій виконується у хмарі Firebase, в яких постачальник хмари діє у якості сервера. Під важкими, розуміємо ресурсовитратні за часом очікування користувача операції роботи з БД, а саме додавання профілів керівників та їх фотографій.

4.3.1 Діаграма класів

Діаграму класів, яка відображає взаємодію React класів, що описують взаємодію між архітектурними рівнями, наведено у графічному матеріалі. На діаграмі зображені наступні класи:

- <App> - основний React клас для оголошення Redux store та залежності Firebase Storage;
- <ParamsForm> - React клас, який використовується класами TeacherForm та StudentForm для взаємодії користувача і візуалізованої форми вибору параметрів студента або керівника;
- <TeacherForm> - React клас-обгортка для створення візуалізованої форми додавання нового керівника до Firebase Realtime Database;
- <StudentForm> - React клас-обгортка для створення візуалізованої форми задання параметрів для отримання рекомендацій щодо керівника;
- <TeacherPage> - React клас-обгортка для класу <TeacherForm>;
- <StudentPage> - React клас-обгортка для класу <StudentForm>;
- <TeacherList> - React клас для динамічного відображення профілів керівників;
- <Firebase> - клас для взаємодії з Firebase Realtime Database та Firebase Storage;

4.3.2 Діаграма послідовності

Діаграму послідовності наведено у графічному матеріалі, вона описує процес отримання рекомендації щодо керівника. На ній відображено послідовність взаємодії користувача та архітектурних рівнів програмного продукту.

4.3.3 Специфікація функцій

Опис основних React/Redux функцій наведено у таблиці 4.1.

Таблиця 4.1 – Специфікація функцій

Назва функції	Примітка
1	2
pushCurrentTeacher: (teacher: any) => (dispatch: any) => void	Асинхронна функція створення об'єкта action для створення запису нового керівника у БД.
setCurrentTeacher: (teacher: any)	Функція створення об'єкта action для додання керівника до локального сховища. Виконується у разі повернення успішного результату функції pushCurrentTeacher
setCurrentStudent: (student: any)	Функція створення об'єкта action для додання параметрів студента до локального дерева стану
sortTeachersByRatings: (studentFilter: any) => (dispatch: any, getState: any) => Promise<void>	Функція створення об'єкта action для отримання відсортованого списку керівників за показниками метрик
setSortedTeachers: (sortedTeachers: any)	Функція створення об'єкта action для отримання відсортованого списку керівників за показниками метрик. Виконується у разі повернення успішного результату функції sortTeachersByRatings

Продовження таблиці 4.1

1	2
currentStudent (state = initialState, action)	Функція reducer, яка приймає поточний стан та дію функції setCurrentStudent.
currentTeacher (state = initialState, action)	Функція reducer, яка приймає поточний стан та дію функції setCurrentTeacher
teachers(state = initialState, action)	Функція reducer, яка приймає поточний стан та дію функції setSortedTeachers
rootReducer	Функція комбінування reducer функцій.
euclidianDistance: (teacher: any, student?: any)	Функція розрахунку відстані Евкліда між об'єктами керівника та студента
manhattanDistance: (teacher: any, student?: any)	Функція розрахунку Манхеттен відстані між об'єктами керівника та студента
jaccardSimilarity: (teacher: any, student?: any)	Функція розрахунку коефіцієнта подібності Жаккара між об'єктами керівника та студента
ParamsForm(props: any)	Компонент відображення форми заповнення параметрів, який повторно використовується у компонентах StudentForm та TeacherForm

Продовження таблиці 4.1

1	2
StudentForm()	Функція створення компонента відображення форми заповнення параметрів студента
TeacherForm()	Функція створення компонента відображення форми заповнення параметрів керівника
TeachersList()	Функція створення компонента відображення списку профілів керівників
TeachersListItem()	Функція створення компонента відображення профілю керівників

Висновок до розділу

У розділі детально описані основні засоби розробки системи та наведені основні переваги кожного засобу. Описані вимоги до технічного забезпечення, які потрібні для використання програмного продукту. Сформульовані вимоги до технічного забезпечення включають вимоги до пристрою, операційної системи та веб-браузеру у якому буде відкрито веб-застосування.

У розділі наведено діаграму класів та наведено детальну специфікацію функцій. Для процесу отримання рекомендації щодо керівника наведено діаграму послідовності.

5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

5.1 Керівництво користувача

Перегляд керівників

Веб-застосування надає можливість студенту переглядати інформацію про керівників та фото, без додаткових очікувань завантажень сторінок. Уся взаємодія та зміни у інтерфейсі відбуваються ніби ми користуємося не веб-сайтом, а застосуванням для десктопу. Для відкриття веб-сайту необхідно відкрити <http://localhost:3000>. Перегляд профілів користувачів зображено рисунку 5.1.

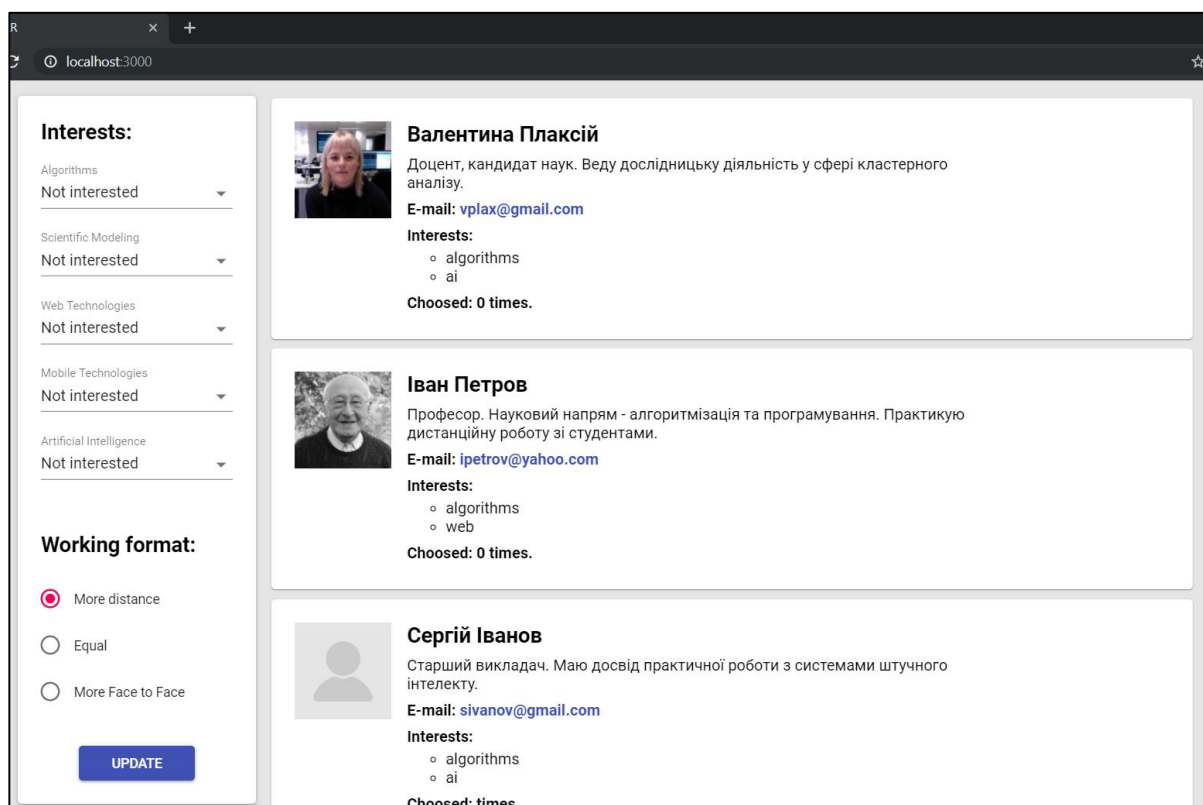


Рисунок 5.1 – Перегляд профілів викладачів

Отримання рекомендованого викладача

Для того, щоб отримати рекомендованого керівника необхідно по кожній з наданих сфер обрати рівень компетентності:

- зацікавлений (Interested);
- не зацікавлений (Not interested);
- має досвід (Experienced).

Для цього треба натиснути на випадаючий список з назвою відповідної наукової сфери. Приклад наведено на рисунку 5.2.

Interests:

Algorithms
Not interested ▼

Not interested
Interested
Experienced

Not interested ▼

Artificial Intelligence
Not interested ▼

Рисунок 5.2 – Вибір рівня компетентності

Для вибору формату зустрічей з керівником треба обрати один з варіантів радіо групи:

- більш дистанційний (More distance);
- посередній (Equal);
- переважно очний (More face to face).

Вибір зображено на рисунку 5.3.

Рисунок 5.3 – Вибір формату зустрічей

Після натиснення кнопки Update отримаємо оновлений список викладачів з розрахованими метриками подібності. Зеленим підсвічено рекомендованого керівника. Результат дії показано на рисунку 5.4

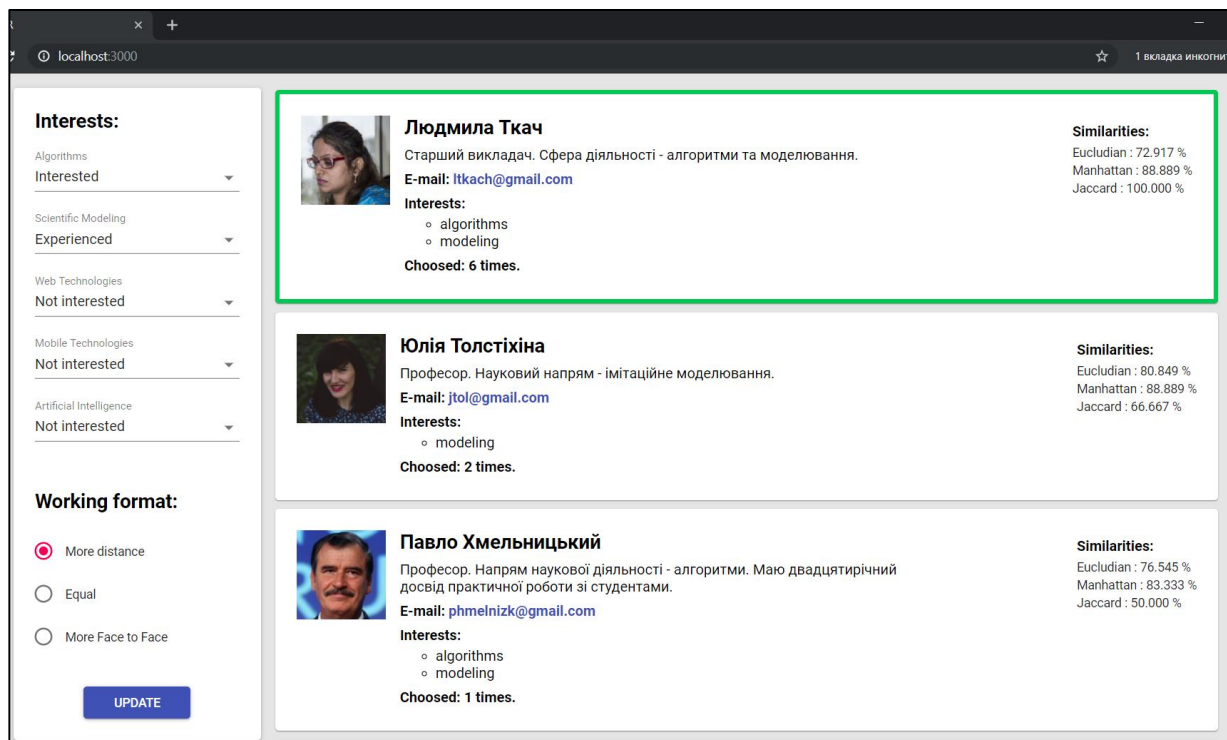


Рисунок 5.4 – Виведення рекомендованого керівника

Додавання нового керівника адміністратором

Для додавання нового керівника адміністратору необхідно перейти на сторінку <http://localhost:3000/teacher-page>. Після цього відкриється сторінка додання нового керівника, яка містить необхідні для заповнення поля ім'я, прізвища, email, деталей про керівника. Результат дії показано на рисунку 5.6.

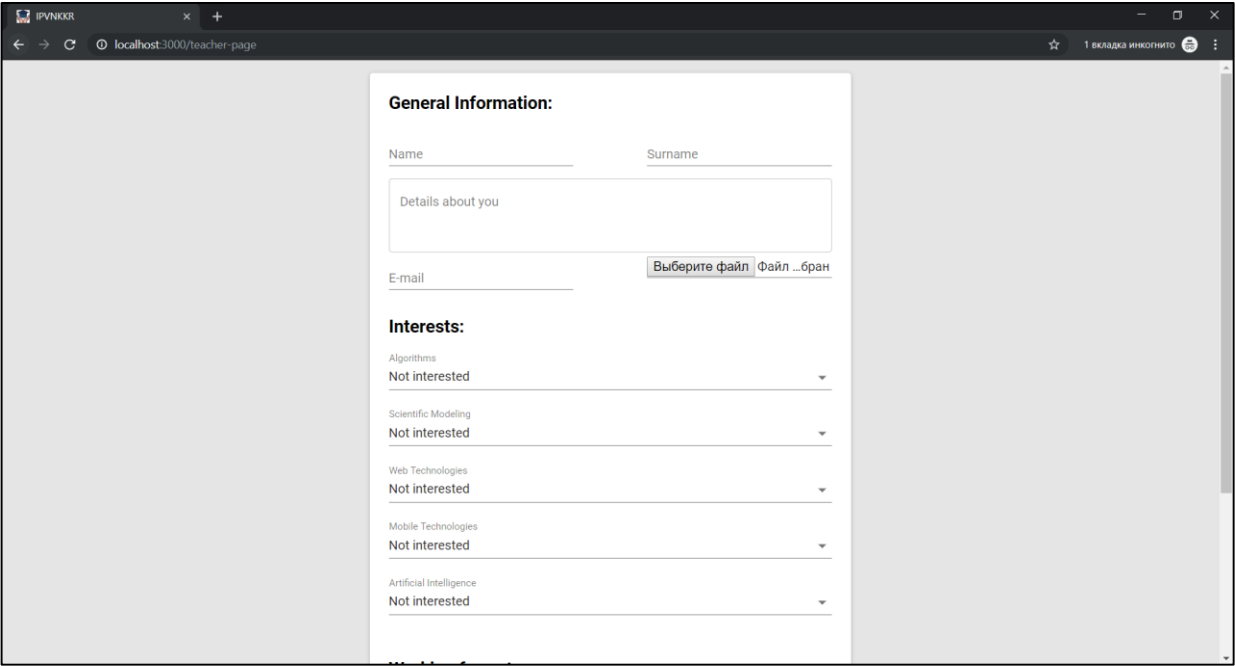


Рисунок 5.6 – Сторінка створення профілю керівника

Для того щоб додати нового керівника необхідно ввести у відповідні поля його ім'я, прізвище, e-mail. Крім цього, треба заповнити текстове поле Про себе. Також слід визначити рівень компетенції керівника вибравши один з варіантів випадаючого списку, обрати формат зустрічей та завантажити фото. Процес завантаження фото зображено на рисунку 5.7.

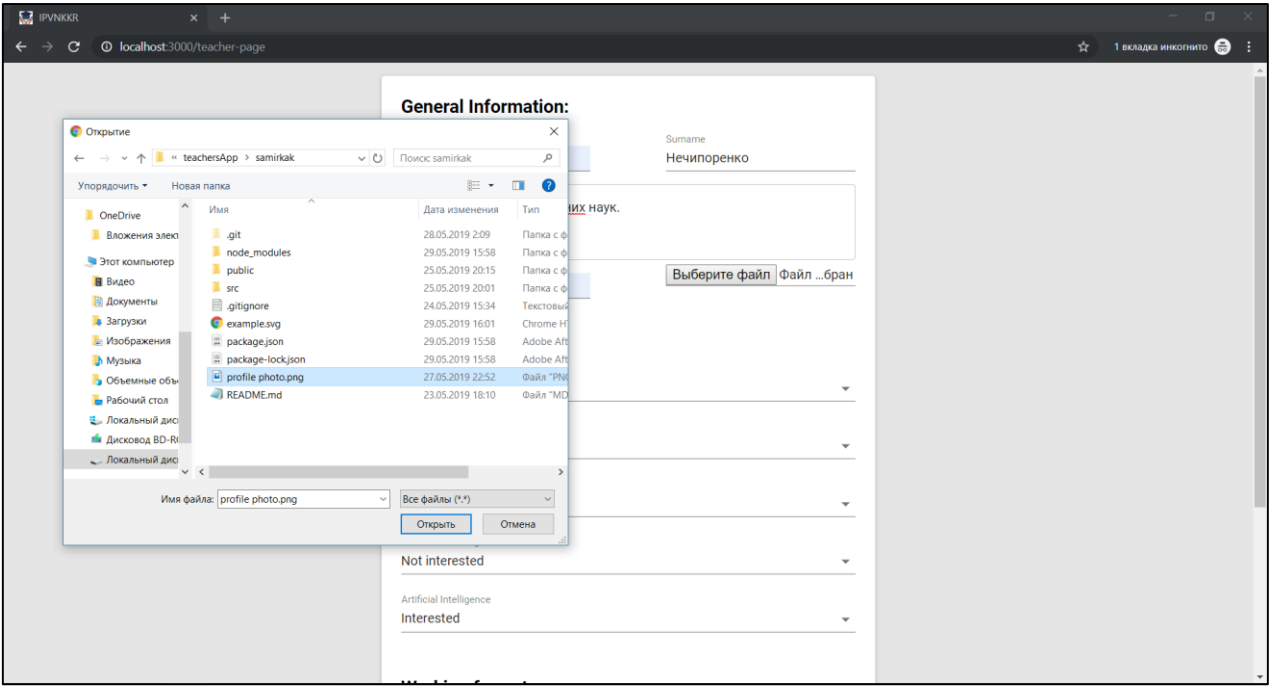


Рисунок 5.7 – Завантаження фото

Після заповнення усіх полів необхідно натиснути на кнопку «Add».
Приклад заповнення показано на рисунку 5.8.

Interests:

Algorithms
Experienced

Scientific Modeling
Not interested

Web Technologies
Experienced

Mobile Technologies
Not interested

Artificial Intelligence
Interested

Working format:

☐ More distance

☐ Equal

☒ More Face to Face

ADD

Рисунок 5.8 – Створення профілю керівника

Якщо залишити будь-яке поле пустим, то у випадку натиснення кнопки Add, виведеться помилка. Приклад валідації зображено на рисунку 5.9.

General Information:

Name
This field is required

Surname
This field is required

Details about you
This field is required

E-mail
This field is required

Выберите файл Файл ...бран

Рисунок 5.9 – Створення профілю керівника

5.2 Випробування програмного продукту

5.2.1 Мета випробувань

Метою випробувань являється перевірка відповідності функцій веб-застосування інформаційної підтримки вибору наукового керівника кваліфікаційної роботи вимогам технічного завдання.

5.2.2 Загальні положення

Випробування проводяться на основі наступних документів:

- ГОСТ 34.603-92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

5.2.3 Результати випробувань

Таблиця 5.1 – Введення довідкової інформації про керівника

Мета тесту:	Перевірка функції «Введення довідкової інформації про керівника»
Початковий стан веб-застосування:	Відкрита сторінка додання нового керівника
Вхідні дані:	Ім'я, Прізвище, Email, текст керівника Про Себе
Схема проведення тесту:	Ввести в поле Name ім'я керівника. Ввести в поле Surname прізвище керівника. Ввести в поле Email електронну адресу керівника. Ввести опис керівника Про Себе в поле Details about you
Очікуваний результат:	Дані у полях заповнені. Сторінка додання нового керівника залишається відкритою

Продовження таблиці 5.1

Стан веб-застосування після проведення вибробувань:	Дані у полях заповнені. Сторінка додання нового керівника залишається відкритою
---	---

Таблиця 5.2 – Визначення області зацікавленості керівника

Мета тесту:	Перевірка функції «Визначення області зацікавленості керівника у науково-технічних сферах»
Початковий стан веб-застосування:	Відкрита сторінка додання нового керівника
Вхідні дані:	Рівень компетентності з трьох варіантів (Interested, Not Interested, Experienced) по кожній з сфер знань (Algorithms, Scientific Modeling, Web Technologies, Mobile Technologies, Artificial Intelligence)
Схема проведення тесту:	Обрати рівень компетентності з Algorithms. Обрати рівень компетентності з Scientific Modeling. Обрати рівень компетентності з Web Technologies. Обрати рівень компетентності з Mobile Technologies. Обрати рівень компетентності з Artificial Intelligence
Очікуваний результат:	Обраний рівень компетентності з трьох варіантів (Interested, Not Interested, Experienced) по кожній з сфер знань (Algorithms, Scientific Modeling, Web Technologies, Mobile Technologies, Artificial Intelligence). Сторінка залишається відкритою

Продовження таблиці 5.2

Стан веб-застосування після проведення вибробувань:	Дані у поля заповнені. Сторінка додання нового керівника залишається відкритою
---	--

Таблиця 5.3 – Задання зручного для керівника режиму робочого процесу

Мета тесту:	Перевірка функції «Задання зручного для керівника режиму робочого процесу»
Початковий стан веб-застосування:	Відкрита сторінка додання нового керівника
Вхідні дані:	Один з трьох форматів робочого процесу (More Distance, Equal, More Face to Face)
Схема проведення тесту:	Обрати формат робочого процесу
Очікуваний результат:	Обраний рівень компетентності з трьох варіантів (Interested, Not Interested, Experienced) по кожній з сфер знань (Algorithms, Scientific Modeling, Web Technologies, Mobile Technologies, Artificial Intelligence) формат робочого процесу. Сторінка залишається відкритою
Стан веб-застосування після проведення вибробувань:	Дані у поля заповнені. Сторінка додання нового керівника залишається відкритою

Таблиця 5.4 – Перегляд профілів керівників

Мета тесту:	Перевірка функції «Перегляд профілів керівників»
Початковий стан веб-застосування:	Профілі керівників не завантажені
Вхідні дані:	
Схема проведення тесту:	Відкрити сторінку перегляду профілів керівників
Очікуваний результат:	Відкрита сторінка перегляду профілів керівників. Профілі керівників завантажені і доступні для перегляду.
Стан веб-застосування після проведення вибробувань:	Відкрита сторінка перегляду профілів керівників. Профілі керівників завантажені і доступні для перегляду.

Таблиця 5.5 – Отримання рекомендованого керівника

Мета тесту:	Перевірка функції «Отримання рекомендованого керівника»
Початковий стан веб-застосування:	Відкрита сторінка перегляду керівників
Вхідні дані:	Обрати рівень компетентності з Algorithms. Обрати рівень компетентності з Scientific Modeling. Обрати рівень компетентності з Web Technologies. Обрати рівень компетентності з Mobile Technologies. Обрати рівень компетентності з Artificial Intelligence.

Продовження таблиці 5.6

Схема проведення тесту:	Обрати рівень компетентності з Algorithms. Обрати рівень компетентності з Scientific Modeling. Обрати рівень компетентності з Web Technologies. Обрати рівень компетентності з Mobile Technologies. Обрати рівень компетентності з Artificial Intelligence. Натиснути кнопку Update.
Очікуваний результат:	Відкрита сторінка перегляду профілів керівників. Рекомендованого керівника підсвічено зеленим.
Стан веб-застосування після проведення випробувань.	Відкрита сторінка перегляду профілів керівників. Рекомендованого керівника підсвічено зеленим.

Таблиця 5.7 – Завантаження фото керівника

Мета тесту:	Перевірка функції «Додавання фото до профілю керівника»
Початковий стан веб-застосування	Відкрита сторінка внесення інформації про керівника
Вхідні дані:	Фотографія керівника у форматі png, jpeg, gif.
Схема проведення тесту:	Натиснути на кнопку «Обрати файл». Обрати файл форматів png, jpeg, gif. Підтвердити вибір
Очікуваний результат:	Фото керівника додано до профілю і є доступним для перегляду.
Стан веб-застосування після проведення випробувань.	Фото керівника додано до профілю і є доступним для перегляду.

Висновок до розділу

У даному розділі наведено детальне керівництво користувача по роботі з веб застосуванням. Наведено приклади зі скріншотами використання основних функцій, таких як: введення адміністратором довідкової інформації та визначення області зацікавленості керівника, задання зручного для керівника режиму робочого процесу, завантаження фото керівника, перегляд студентом профілів керівників, отримання студентом рекомендованого керівника. Основні функції протестовано та виявлено відповідність заявленим вимогам.

					ДП ІС-5110.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

ЗАГАЛЬНІ ВИСНОВКИ

В дипломному проекті було детально розглянуто та проаналізовано процес вибору керівника кваліфікаційної роботи та виявлено, що наразі він відбувається по різному і не завжди приводить до бажаних результатів. Процес інформаційної підтримки вибору наукового керівника може бути покращений, шляхом створення програмного продукту, який дозволить студенту зробити усвідомлений вибір керівника кваліфікаційної роботи. Це збільшить вмотивованість до підвищення якості виконання, як з боку студента, так і керівника.

В процесі пошуку аналогів було зроблено висновок про відсутність схожих програмних рішень, а отже сформовано основні цілі, задачі та мету розробки. В ході проектування програмного продукту інформаційної підтримки було детально розглянуто перелік вхідних даних, які надходять від студента та керівників. Описано вихідні дані та надано відповідні приклади їх виведення.

На основі інформації, отриманої в результаті аналізу процесу вибору керівника було сформовано змістовну постановку задачі, яка полягає у знаходженні рекомендації студенту щодо керівника, сформовано математичну постановку задачі вибору, описано можливі методи розв'язку та розроблено алгоритм розв'язання.

Для розробки веб-застосування було обрано сучасний архітектурний шаблон React/Redux для організації динамічної взаємодії користувача і веб-сайту з використанням хмарних технологій Firebase, які надають можливість ефективної взаємодії з Firebase Realtime Database та Firebase Storage, обґрунтовано основні переваги такого підходу.

Проведено функціональне тестування програмного продукту, розроблено детальну інструкцію користувача зі скріншотами.

					ДП ІС-5110.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

ПЕРЕЛІК ПОСИЛАНЬ

1. Firebase Realtime Database – [Електронний ресурс] : [Веб-сайт]. – Електронні дані – 2019 Google LLC – Режим доступу: <https://firebase.google.com/docs/database/>
2. Что такое NoSQL? – [Електронний ресурс] : [Веб-сайт]. – Електронні дані – 2019 Amazon – Режим доступу: <https://aws.amazon.com/ru/nosql/>
3. Имитационное моделирование в среде GPSS / Томашевский В., Жданова Е. – М.:Бестселлер, 2003. – 416 с
4. Методичні вказівки до виконання дипломного проекту (роботи) першого(бакалаврського) рівня вищої освіти «Бакалавр». [Текст] / Уклад.: М.О.Сперкач – К.: НТУУ «КПІ», 2016. – 53 с.
5. Finding groups in data: An introduction to cluster analysis / Sewell, Grandville, and P. J. Rousseau – Wiley-Interscience – 2005. – 342 с.
6. Five most popular similarity measures - [Електронний ресурс] : [Веб-сайт] // Режим доступу: <https://dataaspirant.com/2015/04/11/five-most-popular-similarity-measures-implementation-in-python/>
7. Офіційний сайт мови JavaScript – [Електронний ресурс] // Режим доступу: <https://www.javascript.com/>
8. React JS - [Електронний ресурс] : [Веб-сайт]. – Електронні дані – 2019 Facebook Inc.– Режим доступу: <https://reactjs.org/>
9. What If React Was Really Only The V in MVC? - [Електронний ресурс] : [Веб-сайт] // Режим доступу: <https://medium.com/javascript-inside/what-if-react-was-really-only-the-v-in-mvc-5854fd6f601c>
10. Redux [Електронний ресурс] : [Веб-сайт]. – Електронні дані – 2019 Dan Abramov and the Redux documentation authors – Режим доступу: <https://redux.js.org/>

11. The best way to architect your Redux app – [Електронний ресурс] :
[Веб-сайт] //Режим доступу: <https://www.freecodecamp.org/news/the-best-way-to-architect-your-redux-app-ad9bd16c8e2d/>

					ДП ІС-5110.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

ДОДАТОК А

Тексти програмного коду

Інформаційна підтримка вибору наукового керівника кваліфікаційної роботи

(Найменування програми)

DVD-R

(Вид носія даних)

30 арк, 271 Мб

(Обсяг програми Мб)

Київ – 2019 року

					ДП ІС-5110.1181-с.ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

```

import {
  SET_CURRENT_TEACHER,
  SET_CURRENT_STUDENT,
  SET_TEACHERS,
  SET_SORTED_TEACHERS,
  OPEN_MODAL,
  CLOSE_MODAL
} from './actionTypes';

import { firebase } from "../App"

import { calculateRatings } from './utils/ratings'

export const pushCurrentTeacher = (teacher) => {
  return (dispatch) => {
    const key = firebase.teachers().push().key;
    const newTeacher = {
      "name": "",
      "surname": "",
      "details": "",
      "email": "",
      "interests": [],
      "workingFormat": 0,
      "choosed": 0
    };
    let photo=null;

    teacher.forEach(element => {
      switch (element.name) {
        case 'name':
          newTeacher[element.name] = element.value;
          break;

```

```

case 'surname':

    newTeacher[element.name] = element.value;

    break;

case 'about':

    newTeacher['details'] = element.value;

    break;

case 'email':

    newTeacher['email'] = element.value;

    break;

case 'workingFormat':

    newTeacher['workingFormat'] = element.value;

    break;

case 'photo':

    photo = element.value;

    break;

default:

    newTeacher['interests'].push({ 'name':element.name, 'value':element.value })

}

});

firebase.profilePictures().child(photo.name).put(photo)

.then((res) => {

    return firebase.profilePictures().child(photo.name).getDownloadURL();

})

.then((url) => {

    newTeacher['imageURL'] = url;

})

.then(() => {

    return firebase.teacher(key).set(newTeacher) //PUSHING TO DATABASE

```

```
    })
    .then(() => {
      dispatch(setCurrentTeacher(newTeacher))
    })
    .catch((error) => {
      dispatch(openModal());
    });
  }
}

export const setCurrentTeacher = (teacher) => {
  return {
    type: SET_CURRENT_TEACHER,
    teacher: {...teacher, id: 0}
  }
};

export const setCurrentStudent = (student) => ({
  type: SET_CURRENT_STUDENT,
  student: student
});

export const setTeachers = (teachers) => {
  const keys = Object.keys(teachers);
  const teachersWithIds = Object.values(teachers).map((teacher,index) =>
  {
    return {...teacher, id: keys[index]}
  }
  )
  return{
    type: SET_TEACHERS,
    teachers: teachersWithIds
  }
```

```

    }
  };

  export const setSortedTeachers = (sortedTeachers) => ({
    type: SET_SORTED_TEACHERS,
    teachers: sortedTeachers
  });

  export const sortTeachersByRatings = (studentFilter) => {
    return (dispatch, getState) => {
      const teachers = getState().teachers.list;

      const teachersWithRatings = calculateRatings(teachers, studentFilter);

      console.log(teachersWithRatings);

      teachersWithRatings[0].choosed++;

      return firebase.teacher(teachersWithRatings[0].id).update({ choosed:
teachersWithRatings[0].choosed })

      .then(() => {
        dispatch(setSortedTeachers(teachersWithRatings))
      })

      .catch((error) => {
        console.log(error);
      });
    }
  }

  export const openModal = () => ({
    type: OPEN_MODAL,
    isOpened: true
  })

  export const closeModal = () => ({
    type: CLOSE_MODAL,
    isOpened: false
  })

```

					ДП ІС-5110.1181-с.ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

```

import app from 'firebase/app';
import 'firebase/database';
import 'firebase/storage';
const config = {
  apiKey: "AIzaSyBnozU7bIQyHY9JQxvk-W3iQuEhDqxFck8",
  authDomain: "teachersapp-2ee85.firebaseio.com",
  databaseURL: "https://teachersapp-2ee85.firebaseio.com",
  projectId: "teachersapp-2ee85",
  storageBucket: "teachersapp-2ee85.appspot.com",
  messagingSenderId: "131476544607",
  appId: "1:131476544607:web:9b442b8dd3a326e7"
};
class Firebase {
  constructor() {
    app.initializeApp(config);
    this.db = app.database();
    this.storage = app.storage();
    console.log(this.storage)
  }
  teacher = uid => this.db.ref(`teachers/${uid}`);
  teachers = () => this.db.ref('teachers');
  profilePictures = () => this.storage.ref('profilePictures');
}
export default Firebase;
import React, { Component } from 'react';
import Grid from '@material-ui/core/Grid';
import InputLabel from "@material-ui/core/InputLabel";
import MenuItem from "@material-ui/core/MenuItem";

```

```

import Select from "@material-ui/core/Select";
import Radio from "@material-ui/core/Radio";
import RadioGroup from "@material-ui/core/RadioGroup";
import FormControlLabel from "@material-ui/core/FormControlLabel";
import FormControl from "@material-ui/core/FormControl";
import Paper from '@material-ui/core/Paper';
import Typography from '@material-ui/core/Typography';
import styles from './ParamsForm.module.css';

import { Button, Input } from '@material-ui/core';
import { ValidatorForm, TextValidator } from 'react-material-ui-form-validator';

export default class ParamsForm extends Component {
  constructor(props) {
    super(props)
    this.state = {
      params: [].concat(...this.props.config.groups.map(group =>
group.controls.map(control => ({
      name: control.name,
      options: control.options,
      type: control.type,
      value: control.type === 'select' || control.type === 'radio' ? 0 : "
      }))))
    }

    this.renderSelect = this.renderSelect.bind(this);
  }

  handleChange = event => {
    const { value, name } = event.target;
    const paramsState = this.state.params.map((param) => (
      param.name === name ?

```



```
        {
            ...param,
            value: param.type === 'radio' ? +value : value
        } :
        param
    )
);
this.setState({ params:paramsState });
}
renderGroup(group){
    return (
<div key={ group.name } className={ styles.group }>
    {
        group.name==='Button'?
        null:
        <h2 variant="h6" className={ styles.groupTitle }>{ group.name }</h2>
    }
    <Grid container justify={ group.name==='Button'? "center" : 'space-between' }>
    {
        group.controls.map(
            control =>
                this.renderControl(control)
        )
    }
    </Grid>
</div>
)

    }

    renderControl(control){
```

```

switch (control.type) {
    case 'select':
        return this.renderSelect(control);
    case 'input':
        if (control.name==='photo') {
            return this.renderImageInput(control);
        } else {
            return this.renderInput(control);
        }
    case 'radio':
        return this.renderRadio(control);
    case 'button':
        return this.renderButton(control);
    case 'textfield':
        return this.renderTextField(control);
    default:
        return null;
}

renderSelect(select){
    const controlState = this.state.params.find((ctl) => ctl.name === select.name);
    return (
        <FormControl fullWidth key={select.name} >
            <InputLabel >{select.label}</InputLabel>
            <Select
                value={controlState.value}
                onChange={this.handleChange}
                inputProps={{ name: select.name, id: select.name }}
                renderValue={() => select.options.find((opt) => opt.value === controlState.value).label}
            >

```

```
        className={styles.select}

        >

            {select.options.map((option) =>

                (<MenuItem

                    value={option.value}

                    key={option.value}>{option.label}

                </MenuItem>)

            )}

        </Select>

    </FormControl>

)

}
```

```
renderTextField(textField){

    const { isValidated } = this.props;

    const controlState = this.state.params.find((ctl) => ctl.name === textField.name);

    return (

        <div key={textField.name} className={styles.textField} >

            <TextValidator

                name={textField.name}

                label={textField.label}

                value={controlState.value}

                onChange={this.handleChange}

                multiline

                fullWidth

                placeholder='About'

                variant="outlined"

                rows={3}

                rowsMax={3}

            />

        </div>

    );

}
```

```
        validators={isValidated ? ['required'] : null}
        errorMessages={['This field is required']}
      />
    </div>
  )
}

handleImageChange = (e) => {
  const file = e.target.files[0];
  const paramsState = this.state.params.map((param) => (
    param.name === 'photo' ?
      {
        ...param,
        value: file
      } :
      param
  ));
  this.setState({ params: paramsState });
}

renderImageInput(input){
  const { isValidated } = this.props;
  const controlState = this.state.params.find((ctl) => ctl.name === input.name);
  return (
    <Grid item xs={5} key={input.name}>
      <FormControl fullWidth>
        <Input
          label={input.label}

```

					ДП ІС-5110.1181-с.ПЗ	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    onChange={this.handleImageChange}

    type='file'

    accept="image/*"

    inputProps={{ name: input.name, id: input.name }}
  />

    </FormControl>

  </Grid>

)

}

renderInput(input){
const { isValidated } = this.props;

    const controlState = this.state.params.find((ctl) => ctl.name === input.name);

    return (

      <Grid item xs={5} key={input.name}>

        <FormControl fullWidth>

          <TextValidator

            label={input.label}

              value={controlState.value}

            onChange={this.handleChange}

            type={input.name === 'photo'? 'file': 'text'}

            accept="image/*"

            inputProps={{ name: input.name, id: input.name }}

            validators={isValidated ? ['required'] : null}

            errorMessages={['This field is required']}

          />

        </FormControl>

      </Grid>

    )

  }

```

```
renderRadio(radioGroup){  
    const controlState = this.state.params.find((ctl) => ctl.name ===  
radioGroup.name);  
    return (  
        <FormControl key={radioGroup.name}>  
            <RadioGroup  
                name={radioGroup.name}  
                value={String(controlState.value)}  
                onChange={this.handleChange}  
            >  
                {  
                    radioGroup.options.map((option) =>  
                        (<FormControlLabel  
                            key={option.value}  
                            value={String(option.value)}  
                            control={<Radio />}  
                            label={option.label}/>))  
                }  
            </RadioGroup>  
        </FormControl>  
    )  
}  
  
renderButton(button){  
    return (  
        <Button  
            variant="contained"  
            color="primary"  
            key={button.value}  
            type='submit'  
            className={styles.button}
```

Змн.	Арк.	№ докум.	Підпис	Дата

```
>

{
  button.value
}
</Button>
)
}

render() {
const { config, onSubmit } = this.props;
const params = this.state.params.filter((param) =>
(
  typeof param.name !== 'undefined'
  &&
  param.name !== null
)
)

return (

<ValidatorForm
  className={styles.main}
  onSubmit={() => onSubmit(params)}
>

      <Paper className={styles.paper}>
        {config.groups.map((group) => this.renderGroup(group))}
      </Paper>
    </ValidatorForm>
  )
}

import React, { Component } from 'react';
```

```

import ParamsForm from '../ParamsForm/ParamsForm';
import studentFormConfig from './studentFormConfig';
import { connect } from "react-redux";
import { bindActionCreators } from 'redux';
import { sortTeachersByRatings } from '../actions/actionCreator'
import styles from './StudentForm.module.css';
class StudentForm extends Component {
  render () {
    return (
      <div className={styles.studentForm}>
        <ParamsForm
          config={studentFormConfig}
          onSubmit={this.props.sortTeachersByRatings}
        />
      </div>
    )
  }
}

const mapDispatchToProps = dispatch => bindActionCreators({ sortTeachersByRatings },
dispatch);
export default connect(null, mapDispatchToProps)(StudentForm);
import React, { Component } from 'react';
import ParamsForm from '../ParamsForm/ParamsForm';
import teacherFormConfig from './teacherFormConfig';
import { connect } from "react-redux";
import { bindActionCreators } from 'redux';
import { pushCurrentTeacher } from '../actions/actionCreator'
import ModalWindow from './ModalWindow/ModalWindow';
class TeacherForm extends Component {

```



```

render () {
  return (
    <div>
      <ParamsForm
        config={teacherFormConfig}
        onSubmit={this.props.pushCurrentTeacher}
        isValidated={true}
      />
      <ModalWindow/>
    </div>
  )
}

const mapDispatchToProps = dispatch => bindActionCreators({ pushCurrentTeacher },
dispatch);

export default connect(null, mapDispatchToProps)(TeacherForm);

export default {
  groups: [
    {
      name: 'General Information:',
      description: 'Information about yourself:',
      controls: [
        {
          type: 'input',
          // options:[{ value:0,label:'More distance'},{ value:1,label:'Equal'},{ value:2,label:'More
Face to Face'}]],
          label: 'Name',
          name: 'name',
        },
        {

```

```

    type: 'input',

    // options:[{ value:0,label:'More distance'},{ value:1,label:'Equal'},{ value:2,label:'More
Face to Face'}]],

    label: 'Surname',

    name: 'surname',

  },

  {

    type: 'textfield',

    // options:[{ value:0,label:'More distance'},{ value:1,label:'Equal'},{ value:2,label:'More
Face to Face'}]],

    label: 'Details about you',

    name: 'about',

  },

  {

    type: 'input',

    // options:[{ value:0,label:'More distance'},{ value:1,label:'Equal'},{ value:2,label:'More
Face to Face'}]],

    label: 'E-mail',

    name: 'email',

  },

  {

    type: 'input',

    // options:[{ value:0,label:'More distance'},{ value:1,label:'Equal'},{ value:2,label:'More
Face to Face'}]],

    label: 'Photo',

    name: 'photo',

  },

],

},

{

  name: 'Interests:',

```

					ДП ІС-5110.1181-с.ПЗ	Арк.
						73
Змн.	Арк.	№ докум.	Підпис	Дата		

description: 'Academic fields you are interested in:',

controls: [

{

type: 'select',

options: [{ value: 0, label: 'Not interested' }, { value: 1, label: 'Interested' }, { value: 2, label: 'Experienced' }],

label: 'Algorithms',

name: 'algorithms'

},

{

type: 'select',

options: [{ value: 0, label: 'Not interested' }, { value: 1, label: 'Interested' }, { value: 2, label: 'Experienced' }],

label: 'Scientific Modeling',

name: 'modeling'

},

{

type: 'select',

options: [{ value: 0, label: 'Not interested' }, { value: 1, label: 'Interested' }, { value: 2, label: 'Experienced' }],

label: 'Web Technologies',

name: 'web'

},

{

type: 'select',

options: [{ value: 0, label: 'Not interested' }, { value: 1, label: 'Interested' }, { value: 2, label: 'Experienced' }],

label: 'Mobile Technologies',

name: 'mobile',

},

{

```
type: 'select',

options: [{ value: 0, label: 'Not interested' }, { value: 1, label: 'Interested' }, { value: 2,
label: 'Experienced' }],

label: 'Artificial Intelligence',

name: 'ai',

}

],

},

{

name: 'Working format:',

description: "",

controls: [

{

type: 'radio',

options: [{ value: 0, label: 'More distance' }, { value: 1, label: 'Equal' }, { value: 2, label:
'More Face to Face' }],

label: 'Meetings format',

name: 'workingFormat',

}

],

},

{

name: 'Button',

controls: [

{

type: 'button',

value: 'Add'

}

]

}
```

					ДП ІС-5110.1181-с.ПЗ	Арк.
						75
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    ],
  }

import React, { Component } from 'react';
import TeacherListItem from '../TeachersListItem/TeacherListItem';
import List from '@material-ui/core/List';
import styles from '../TeachersList.module.css';
import { withFirebase } from '../../components/Firebase/context';
import { connect } from "react-redux";
import { bindActionCreators } from 'redux';
import { setTeachers } from '../../actions/actionCreator'
import { calculateRatings } from '../../utils/ratings'
class TeachersList extends Component {
  componentDidMount() {
    const { firebase , setTeachers } = this.props;

    firebase.teachers().on('value', (teachers) => {
      setTeachers(teachers.val());
    });
  }
  render() {
    const { teachers } = this.props;
    return (
      <List className={styles.container}>
        {
          teachers && teachers.map(teacher => <TeacherListItem teacher={teacher}
key={teacher.name}/>)
        }
      </List>
    )
  }
}

```

```

    }
  }
  const mapStateToProps = state => {
    // const studentFilter = state.currentStudent

    const teachers = state.teachers.sortedList || state.teachers.list;

    // const teachersWithRatings = calculateRatings(teachers, studentFilter);

    return {
      teachers
    }
  }

  const mapDispatchToProps = dispatch => bindActionCreators({ setTeachers }, dispatch);

  export default connect(mapStateToProps, mapDispatchToProps)(withFirebase(TeachersList));
  import {
    SET_CURRENT_STUDENT
  } from '../actions/actionTypes'

  const initialState = { };

  export default (state = initialState, action) => {
    switch (action.type){
      case SET_CURRENT_STUDENT:
        return action.student;
      default:
        return state;
    }
  }

  import {
    SET_CURRENT_TEACHER

```

					ДП ІС-5110.1181-с.ПЗ	Арк.
						77
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    } from '../actions/actionTypes'

const initialState = { };

export default (state = initialState, action) => {
    switch (action.type){
    case SET_CURRENT_TEACHER:
        return action.teacher;
        default:
            return state;
    }
}

import _ from 'lodash'

export const calculateRatings = (teachers, studentFilter) => {
    // console.log(teachers,studentFilter);
    if (Object.keys(studentFilter).length === 0)
        return teachers;
    else {
        const eucludianDistance = function(teacher, student = studentFilter){
            let dist = 0;
            teacher.interests.map( interest => {
                const interestDiff = interest.value - student.find((element) => (interest.name ===
                element.name)).value;
                dist += Math.pow(interestDiff,2);
            }
        )
        const workingFormatDiff = teacher.workingFormat - student.find((element) =>
        (element.name === 'workingFormat')).value;
        dist += Math.pow(workingFormatDiff,2);
        return Math.sqrt(dist);
    }
}

```

```

    }

    const manhattanDistance = function(teacher, student = studentFilter){

        let dist = 0;

        teacher.interests.map( interest => {

            const interestDiff = interest.value - student.find((element) => (interest.name ===
            element.name)).value;

            dist += Math.abs(interestDiff);

        }

    )

    const workingFormatDiff = teacher.workingFormat - student.find((element) =>
    (element.name === 'workingFormat')).value;

    dist += Math.abs(workingFormatDiff,2);

    return dist;

}

const jaccardSimilarity = function(teacher, student = studentFilter){

    const numerator = teacher.interests.reduce(

        (num,curInterest) => {

            const curStudentFilter = student.find((element) => (curInterest.name === element.name));

            return num += Math.min(curInterest.value, curStudentFilter.value);

        }, 0

    );

    const denominator = teacher.interests.reduce(

        (denom,curInterest) => {

            const curStudentFilter = student.find((element) => (curInterest.name === element.name));

            return denom += Math.max(curInterest.value, curStudentFilter.value);

        }, 0

    );

    return (numerator/denominator)

}

```



```

const eucludianMaxValue = 7.38469228349534;

const manhattanMaxValue = 18;

const teachersWithRatings = teachers.map( teacher => {

  return { ...teacher,

    ratings: [

      { name: 'Eucludian', value: (100-(eucludianDistance(teacher)/eucludianMaxValue)*100)

    },

      { name: 'Manhattan', value: (100-
(manhattanDistance(teacher)/manhattanMaxValue)*100) },

      { name: 'Jaccard', value: jaccardSimilarity(teacher)*100 }

    ]

  }

}

return _.sortBy(teachersWithRatings, [function(teacher){

  const averageRating = teacher.ratings.reduce(

    (average,curRating) => {

      return (average += curRating.value)

    }, 0

  );

  return averageRating/teacher.ratings.length;

}]).reverse();

}

};

import React from 'react';

import { Provider } from 'react-redux';

import { Switch, Route } from 'react-router'

import { BrowserRouter as Router } from 'react-router-dom'

import { ConnectedRouter } from 'connected-react-router';

```

					ДП ІС-5110.1181-с.ПЗ	Арк.
						80
Змн.	Арк.	№ докум.	Підпис	Дата		

```

import configureStore from './store';

import { createBrowserHistory } from 'history';

import Firebase, { FirebaseContext } from './components/Firebase';

import StudentPage from './pages/StudentPage/StudentPage.js'
import TeacherPage from './pages/TeacherPage/TeacherPage.js'

export const history = createBrowserHistory();
export const firebase = new Firebase();
export const store = configureStore(history);

function App () {
  return (
    <FirebaseContext.Provider value={firebase}>
      <Provider store={store}>
        <ConnectedRouter history={history}>
          <Router>
            <Switch>
              <Route exact path="/" render={() => (<StudentPage/>)} />
              <Route path="/teacher-page" render={() => (<TeacherPage/>)} />
            </Switch>
          </Router>
        </ConnectedRouter>
      </Provider>
    </FirebaseContext.Provider>
  )
}

export default App

import {
  createStore,

```

```

    applyMiddleware,
    compose
  } from 'redux';
import thunk from 'redux-thunk';
import logger from 'redux-logger'
import rootReducer from './reducers/rootReducer';
import { routerMiddleware } from 'connected-react-router';

export default function configureStore(history) {
  return createStore(
    rootReducer(history), {},
    compose(
      applyMiddleware(
        routerMiddleware(history),
        thunk,
        logger
      )
    )
  );
}

// This optional code is used to register a service worker.
// register() is not called by default.

// This lets the app load faster on subsequent visits in production, and gives
// it offline capabilities. However, it also means that developers (and users)
// will only see deployed updates on subsequent visits to a page, after all the
// existing tabs open on the page have been closed, since previously cached
// resources are updated in the background.

```

					ДП ІС-5110.1181-с.ПЗ	Арк.
						82
Змн.	Арк.	№ докум.	Підпис	Дата		

// To learn more about the benefits of this model and instructions on how to

// opt-in, read <https://bit.ly/CRA-PWA>

```
const isLocalhost = Boolean(
```

```
  window.location.hostname === 'localhost' ||
```

```
  // [::1] is the IPv6 localhost address.
```

```
  window.location.hostname === '[:1]' ||
```

```
  // 127.0.0.1/8 is considered localhost for IPv4.
```

```
  window.location.hostname.match(
```

```
    /^127(?:\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)){3}$/
```

```
)
```

```
);
```

```
export function register(config) {
```

```
  if (process.env.NODE_ENV === 'production' && 'serviceWorker' in navigator) {
```

```
    // The URL constructor is available in all browsers that support SW.
```

```
    const publicUrl = new URL(process.env.PUBLIC_URL, window.location.href);
```

```
    if (publicUrl.origin !== window.location.origin) {
```

```
      // Our service worker won't work if PUBLIC_URL is on a different origin
```

```
      // from what our page is served on. This might happen if a CDN is used to
```

```
      // serve assets; see https://github.com/facebook/create-react-app/issues/2374
```

```
      return;
```

```
    }
```

```
    window.addEventListener('load', () => {
```

```
      const swUrl = `${process.env.PUBLIC_URL}/service-worker.js`;
```

```
      if (isLocalhost) {
```

```
        // This is running on localhost. Let's check if a service worker still exists or not.
```

```

checkValidServiceWorker(swUrl, config);

// Add some additional logging to localhost, pointing developers to the
// service worker/PWA documentation.
navigator.serviceWorker.ready.then(() => {
  console.log(
    'This web app is being served cache-first by a service ' +
    'worker. To learn more, visit https://bit.ly/CRA-PWA'
  );
});
} else {
  // Is not localhost. Just register service worker
  registerValidSW(swUrl, config);
}
});
}
}

function registerValidSW(swUrl, config) {
  navigator.serviceWorker
    .register(swUrl)
    .then(registration => {
      registration.onupdatefound = () => {
        const installingWorker = registration.installing;
        if (installingWorker == null) {
          return;
        }
        installingWorker.onstatechange = () => {
          if (installingWorker.state === 'installed') {
            if (navigator.serviceWorker.controller) {

```

```

// At this point, the updated precached content has been fetched,
// but the previous service worker will still serve the older
// content until all client tabs are closed.

console.log(
  'New content is available and will be used when all ' +
  'tabs for this page are closed. See https://bit.ly/CRA-PWA.'
);

// Execute callback
if (config && config.onUpdate) {
  config.onUpdate(registration);
}
} else {
  // At this point, everything has been precached.
  // It's the perfect time to display a
  // "Content is cached for offline use." message.
  console.log('Content is cached for offline use.');
```

```

// Execute callback
if (config && config.onSuccess) {
  config.onSuccess(registration);
}
}
}
};
};
})
.catch(error => {
  console.error('Error during service worker registration:', error);
});

```

```

}

function checkValidServiceWorker(swUrl, config) {

  // Check if the service worker can be found. If it can't reload the page.

  fetch(swUrl)

    .then(response => {

      // Ensure service worker exists, and that we really are getting a JS file.

      const contentType = response.headers.get('content-type');

      if (

        response.status === 404 ||

        (contentType !== null && contentType.indexOf('javascript') === -1)

      ) {

        // No service worker found. Probably a different app. Reload the page.

        navigator.serviceWorker.ready.then(registration => {

          registration.unregister().then(() => {

            window.location.reload();

          });

        });

      } else {

        // Service worker found. Proceed as normal.

        registerValidSW(swUrl, config);

      }

    })

    .catch(() => {

      console.log(

        'No internet connection found. App is running in offline mode.'

      );

    });

}

export function unregister() {

```

```
if ('serviceWorker' in navigator) {  
  navigator.serviceWorker.ready.then(registration => {  
    registration.unregister();  
  });  
}
```

					ДП ІС-5110.1181-с.ПЗ	Арк.
						87
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

1	ЗАГАЛЬНІ ПОЛОЖЕННЯ	3
1.1	ПОВНЕ НАЙМЕНУВАННЯ СИСТЕМИ ТА ЇЇ УМОВНЕ ПОЗНАЧЕННЯ.....	3
1.2	НАЙМЕНУВАННЯ ОРГАНІЗАЦІЇ-ЗАМОВНИКА ТА УЧАСНИКІВ РОБІТ	3
1.3	ПЕРЕЛІК ДОКУМЕНТІВ, НА ПІДСТАВІ ЯКИХ СТВОРЮЄТЬСЯ СИСТЕМА	3
1.4	ПЛАНОВІ ТЕРМІНИ ПОЧАТКУ І ЗАКІНЧЕННЯ РОБОТИ ЗІ СТВОРЕННЯ СИСТЕМИ .	3
2	ПРИЗНАЧЕННЯ І МЕТА СТВОРЕННЯ ЗАСТОСУВАННЯ	5
2.1	ПРИЗНАЧЕННЯ ЗАСТОСУВАННЯ	5
2.2	ЦІЛІ СТВОРЕННЯ ЗАСТОСУВАННЯ	5
3	ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ.....	5
4	ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	7
4.1	ВИМОГИ ДО ФУНКЦІОНАЛЬНИХ ХАРАКТЕРИСТИК	7
4.2	ВИМОГИ ДО НАДІЙНОСТІ	7
4.3	ВИМОГИ ДО СКЛАДУ І ПАРАМЕТРІВ ТЕХНІЧНИХ ЗАСОБІВ.....	7
5	СТАДІЇ І ЕТАПИ РОЗРОБКИ.....	9
6	ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ СИСТЕМИ.....	10
6.1	ВИДИ ВИПРОБУВАНЬ.....	10

					ДП ІС-5110.1181-с.ТЗ			
		Прізвище	Підпис	Дата				
Розроб.		Как С.Р.			Інформаційна підтримка вибору наукового керівника кваліфікаційної роботи	Літ.	Лист	Листів
Перевірів		Тєлишева Т.О.					2	10
Н. кон.		Тєлишева Т.О.				КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51		
Затв.		Павлов О.А.						

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Повне найменування системи та її умовне позначення

Повна назва системи: Інформаційна підтримка вибору наукового керівника кваліфікаційної роботи

1.2 Найменування організації-замовника та організацій-учасників робіт

Генеральним замовником проекту є кафедра «Автоматизованих систем обробки інформації та управління» «КПІ ім. Ігоря Сікорського». Представником замовника Телишева Тамара Олексіївна.

Розробником системи є студент групи ІС-51 факультету інформатики та обчислювальної техніки «КПІ ім. Ігоря Сікорського» Как Самір Рахулович.

1.3 Перелік документів, на підставі яких створюється система

При розробці системи і створення проектно-експлуатаційної документації Виконавець повинен керуватися вимогами наступних нормативних документів:

- ДСТУ 19.201-78. Технічне завдання. Вимоги до змісту і оформлення;
- ДСТУ 34.601-90. Комплекс стандартів на автоматизовані системи. Автоматизовані системи. Стадії створення;
- ДСТУ 34.201-89. Інформаційні технології. Комплекс стандартів на автоматизовані системи. Види, комплексність і позначення документів при створенні автоматизованих систем.

1.4 Планові терміни початку і закінчення роботи зі створення системи

					ДП ІС-5110.1181-с.ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

Плановий термін початку роботи над інформаційною підтримкою вибору керівника кваліфікаційної роботи – 5 лютого 2019 рік.

Плановий термін по закінченню роботи над інформаційною підтримкою вибору наукового керівника кваліфікаційної роботи – не пізніше 1 червня 2019 року.

					ДП ІС-5110.1181-с.ТЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

2 ПРИЗНАЧЕННЯ І МЕТА СТВОРЕННЯ СИСТЕМИ

2.1 Призначення системи

Призначення системи інформаційної підтримки вибору наукового керівника – дозволити студенту зробити усвідомлений вибір керівника кваліфікаційної роботи на основі отримання розширеної інформації щодо керівників та рекомендацій вибору.

2.2 Цілі створення системи

Головною метою розробки є спрощення процесу вибору дипломного керівника.

При вирішенні зазначених задач будуть досягнуті наступні цілі:

- усвідомлений та мотивований пошук керівника;
- впорядкування комфортного робочого процесу для студента і керівника;
- підвищення якості кваліфікаційних робіт внаслідок сумісної мотивованої роботи та зацікавленості обох сторін у подальшому розвитку співпраці;

Для досягнення мети необхідне вирішення наступних задач:

- розробка концепції системи і аналіз існуючих бізнес процесів;
- визначення функцій і форми представлення системи;
- проектування архітектури та розробка програмного забезпечення, бази даних;
- запропонування математичних засад формування підтримки вибору;
- проведення тестування розробленої системи на якість.

3 ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ

Об'єктом автоматизації виступає процес вибору дипломного керівника.

					ДП ІС-5110.1181-с.ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

Для користування системою обов'язковою умовою для користувача є наявність браузера та підключення до мережі Інтернет, адже програмний продукт є веб-застосуванням.

Працювати з веб-застосуванням можна користувачам без попередньої авторизації, користувачі мають доступ до більшості функцій, окрім внесення даних про керівників, що є доступним лише адміністратору.

					ДП ІС-5110.1181-с.ТЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Вимоги до функціональних характеристик

Дана система, повинна задовольняти потреби студентів, які зацікавлені у пошуку керівника кваліфікаційної роботи. Програмне забезпечення має виконувати наступні функції:

- перегляд профілів керівників з інформацією, наданою студенту у доступ;
- отримання рекомендації щодо вибору керівника;
- формування шаблону листа для комунікації з викладачем;
- внесення інформації про керівників до бази даних.

4.2 Вимоги до надійності

Веб-застосування повинно зберігати працездатність і забезпечувати відновлення своїх функцій при виникненні наступних позаштатних ситуацій:

- при помилках в роботі апаратних засобів (крім носіїв даних і програм) відновлення функції програми покладається на хостинг;
- при помилках, пов'язаних з особливістю різноманітних браузерів, існуючих на сьогоднішній день.

Програмний продукт повинен поєднувати надійність та функціональність. У разі виникнення аварійних ситуацій необхідно сповіщати користувача та надавати інструкцію для подальших дій.

4.3 Вимоги до складу і параметрів технічних засобів

Програмний продукт є веб-застосуванням, використання якого не потребує спеціальних знань або кваліфікацій.

Для забезпечення коректної та стабільної роботи необхідне забезпечення виконання усіх наступних вимог:

1. Пристрій:

					ДП ІС-5110.1181-с.ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

- будь-який сучасний пристрій з можливістю доступу до мережі інтернет

2. Операційні системи:

- Windows
- MacOS
- Linux

3. Веб-браузер:

- Google Chrome версії не нижче 74.0.3729.169
- Safari версії не нижче 12.1.1
- Mozilla Firefox версії не нижче 67.0

4. Доступ до мережі Інтернет

					ДП ІС-5110.1181-с.ТЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

5 СТАДІЇ І ЕТАПИ РОЗРОБКИ

Основні етапи виконання робіт з розробки інформаційної підтримки вибору наукового керівника кваліфікаційної роботи.

№ П.№	Назва етапу роботи	Термін виконання етапу	Результат виконання
1.	Підготовка технічного завдання на розробку програмного продукту	07.02.2019	ТЗ підготовлено
2.	Розробка сценарію роботи	12.02.2019	Сценарій розроблено
3.	Технічне проектування – функціональність, модулі, задачі, цілі тощо	20.02.2019	Функціональність, модулі, задачі, цілі
4.	Узгодження з керівником інтерфейсу користувача	02.03.2019	Інтерфейс узгоджено
5.	Розробка інформаційного забезпечення	17.03.2019	Інформаційне забезпечення розроблено
6.	Розробка програмного забезпечення	29.03.2019	Програмне забезпечення розроблено
7.	Налагодження програми	13.04.2019	Налагоджена програма
8.	Тестування програми	27.04.2019	Програма протестована
9.	Здача готового програмного продукту замовнику	12.05.2019	ПЗ готове

6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ СИСТЕМИ

6.1 Види випробувань

Для контролю правильності роботи програмного забезпечення було проведено функціональне тестування. В ході тестування було проведено випробування основного функціоналу системи.

Список проведених тестувань:

- тестування функції введення довідкової інформації про керівника;
- тестування визначення області зацікавленості керівника у науково-технічних сферах;
- тестування задання зручного для керівника режиму робочого процесу;
- тестування перегляду профілів керівників;
- тестування отримання рекомендованого керівника;
- тестування додання фото до профілю керівника.

ЗМІСТ

1	ОБ'ЄКТ ВИПРОБУВАННЯ.....	3
1.1	Найменування програми	3
1.2	Область застосування	3
1.3	Умовне позначення програми.....	3
2	Мета випробувань	4
3	Вимоги до програмного продукту	4
3.1	Вимоги до функціональних характеристик.....	5
3.1.1	Вимоги до складу виконуваних функцій.....	5
4	Вимоги до програмної документації	6
5	Склад і порядок випробувань	7
6	Методи випробувань	7

					ДП ІС-5110.1181-с.ПМВ				
Зм.	Арк.	Прізвище	Підпис	Дата	Інформаційна підтримка вибору науко- вого керівника кваліфікаційної роботи	Літ.	Лист	Листів	
Розроб.		Как С.Р.							
							2	12	
Перевірів.		Тєлишева Т.О.				КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51			
Н. кон.		Тєлишева Т.О.							
Затв.		Павлов О.А.							

1 ОБ'ЄКТ ВИПРОБУВАННЯ

1.1 Найменування програми

Інформаційна підтримка вибору наукового керівника кваліфікаційної роботи.

1.2 Область застосування

Програмний продукт призначений допомогти студенту у процесі пошуку та вибору дипломного керівника шляхом розширення інформаційного поля та надання рекомендацій.

1.3 Умовне позначення програми

Умовне позначення: «IPVNKKR».

					ДП ІС-5110.1181-с.ПМВ	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

2 МЕТА ВИПРОБУВАНЬ

Метою випробувань є перевірка коректної роботи основного функціоналу та забезпечення відповідності вимогам, які були наведені у технічному завданні.

					ДП ІС-5110.1181-с.ПМВ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

3 ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

3.1 Вимоги до функціональних характеристик

1. Програмний продукт повинен надати студенту можливість переглядати профілі керівників з інформацією, наданою йому у доступ.
2. Програмний продукт повинен надати студенту можливість отримання рекомендації щодо вибору керівника.
3. Програмний продукт повинен надати можливість студенту задати зацікавленість науковими напрямками відповідно до визначеної шкали.
4. Програмний продукт повинен надавати студенту можливість задати зручний йому формат зустрічей відповідно до визначеної шкали.
5. Програмний продукт повинен надати можливість студенту сформувати шаблон листа для комунікації з викладачем.
6. Програмний продукт повинен надати можливість адміністратору вносити інформацію про керівників до бази даних.

3.1.1 Вимоги до складу виконуваних функцій

Функціонал повинен забезпечувати можливість виконання перерахованих нижче функцій:

- перегляд профілів керівників;
- отримання рекомендації щодо вибору керівника ;
- внесення даних про керівника до БД.

4 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

Програмна документація має складатися з керівництва користувача та вихідних текстів програмного коду.

					ДП ІС-5110.1181-с.ПМВ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

5 СКЛАД І ПОРЯДОК ВИПРОБУВАНЬ

Етапи випробувань:

- ознайомчий;
- виконавчий.

На ознайомчому етапі проводиться:

- перевірка комплектності програмної документації;
- перевірка комплектності складу технічних і програмних засобів.

Під час виконавчого етапу проводиться:

- перевірка відповідності технічних характеристик системи;
- перевірка ступеню виконання вимог функціонального призначення системи.

Функції, що підлягають перевірці:

- введення довідкової інформації про керівника;
- задання області зацікавленості керівника у науково-технічних сферах;
- задання зручного для керівника режиму робочого процесу;
- перегляд профілів керівників;
- отримання рекомендації щодо керівника;
- додання фото до профілю керівника.

6 МЕТОДИ ВИПРОБУВАНЬ

У процесі тестування була перевірена основна функціональність застосування. У таблицях 6.1 – 6.6 наведено перелік випробувань основних функціональних можливостей.

Таблиця 6.1 – Тест введення довідкової інформації про керівника

Мета тесту:	Перевірка функції «Введення довідкової інформації про керівника»
Початковий стан веб-застосування:	Відкрита сторінка додання нового керівника
Вхідні дані:	Ім'я, Прізвище, Email, текст керівника Про Себе
Схема проведення тесту:	Ввести в поле Name ім'я керівника. Ввести в поле Surname прізвище керівника. Ввести в поле Email електронну адресу керівника. Ввести опис керівника Про Себе в поле Details about you
Очікуваний результат:	Дані у полях заповнені. Сторінка додання нового керівника залишається відкритою
Стан веб-застосування після проведення випробувань:	Дані у полях заповнені. Сторінка додання нового керівника залишається відкритою

Таблиця 6.2 – Тест визначення області зацікавленості керівника

Мета тесту:	Перевірка функції «Задання області зацікавленості керівника у науково-технічних сферах»
Початковий стан веб-застосування:	Відкрита сторінка додання нового керівника

Продовження таблиці 6.2

Вхідні дані:	Рівень компетентності з трьох варіантів (Interested, Not Interested, Experienced) по кожній з сфер знань (Algorithms, Scientific Modeling, Web Technologies, Mobile Technologies, Artificial Intelligence)
Схема проведення тесту:	Обрати рівень компетентності з Algorithms. Обрати рівень компетентності з Scientific Modeling. Обрати рівень компетентності з Web Technologies. Обрати рівень компетентності з Mobile Technologies. Обрати рівень компетентності з Artificial Intelligence
Очікуваний результат:	Обраний рівень компетентності з трьох варіантів (Interested, Not Interested, Experienced) по кожній з сфер знань (Algorithms, Scientific Modeling, Web Technologies, Mobile Technologies, Artificial Intelligence). Сторінка залишається відкритою
Стан веб-застосування після проведення вибробувань:	Дані у поля заповнені. Сторінка додавання нового керівника залишається відкритою

Таблиця 6.3 – Тест задання зручного режиму робочого процесу

Мета тесту:	Перевірка функції «Задання зручного для керівника режиму робочого процесу»
Початковий стан веб-застосування:	Відкрита сторінка додавання нового керівника
Вхідні дані:	Один з трьох форматів робочого процесу (More Distance, Equal, More Face to Face)

Продовження таблиці 6.3

Схема проведення тесту:	Обрати формат робочого процесу
Очікуваний результат:	Обраний рівень компетентності з трьох варіантів (Interested, Not Interested, Experienced) по кожній з сфер знань (Algorithms, Scientific Modeling, Web Technologies, Mobile Technologies, Artificial Intelligence) формат робочого процесу. Сторінка залишається відкритою
Стан веб-застосування після проведення вибробувань:	Дані у поля заповнені. Сторінка додання нового керівника залишається відкритою

Таблиця 6.4 – Тест перегляду профілів керівників

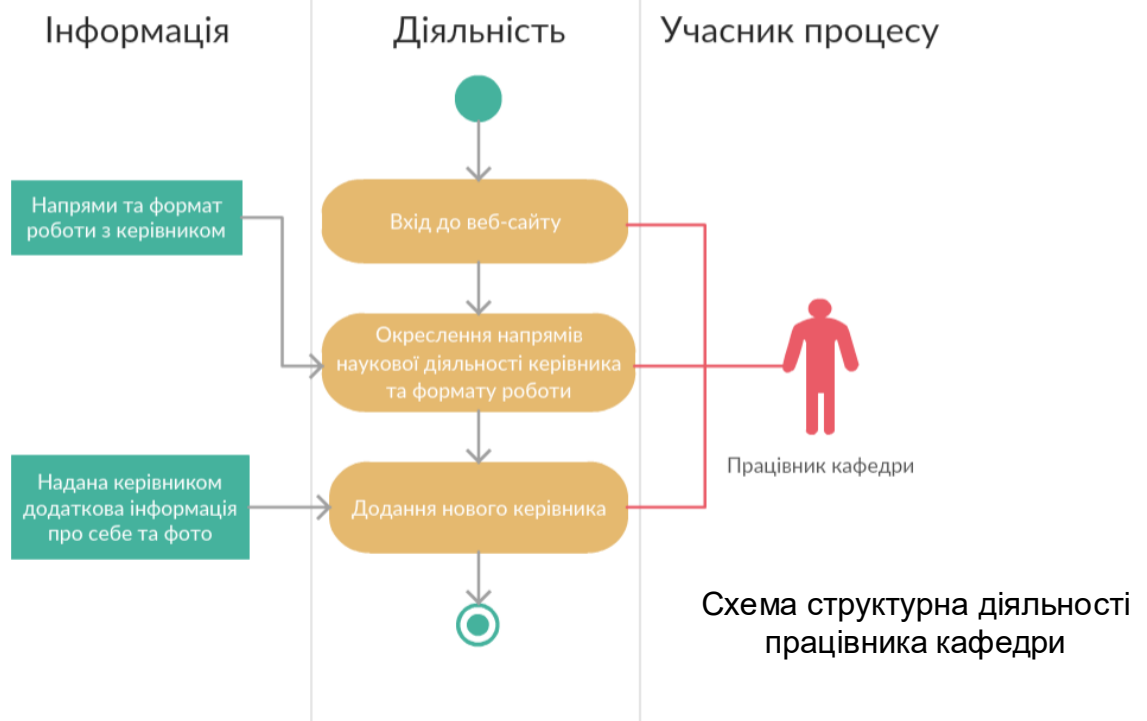
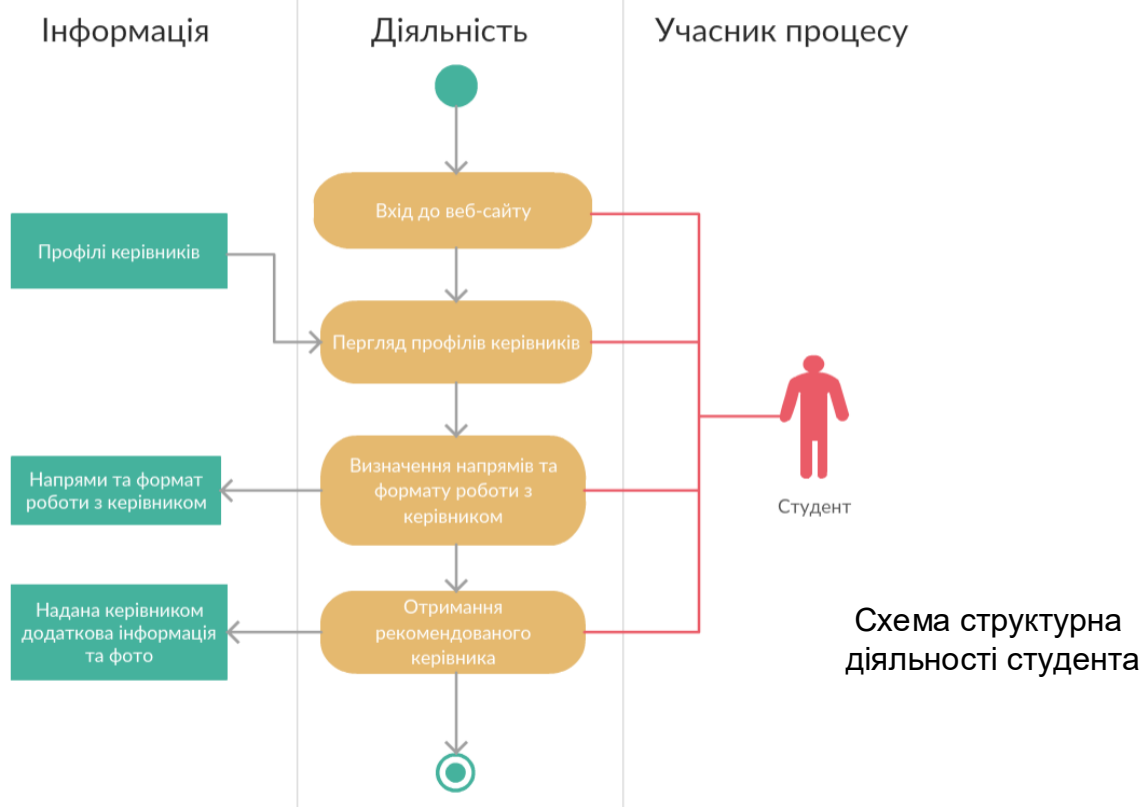
Мета тесту:	Перевірка функції «Перегляд профілів керівників»
Початковий стан веб-застосування:	Профілі керівників не завантажені
Вхідні дані:	
Схема проведення тесту:	Відкрити сторінку перегляду профілів керівників
Очікуваний результат:	Відкрита сторінка перегляду профілів керівників. Профілі керівників завантажені і доступні для перегляду.
Стан веб-застосування після проведення вибробувань:	Відкрита сторінка перегляду профілів керівників. Профілі керівників завантажені і доступні для перегляду.

Таблиця 6.5 – Тест отримання рекомендованого керівника

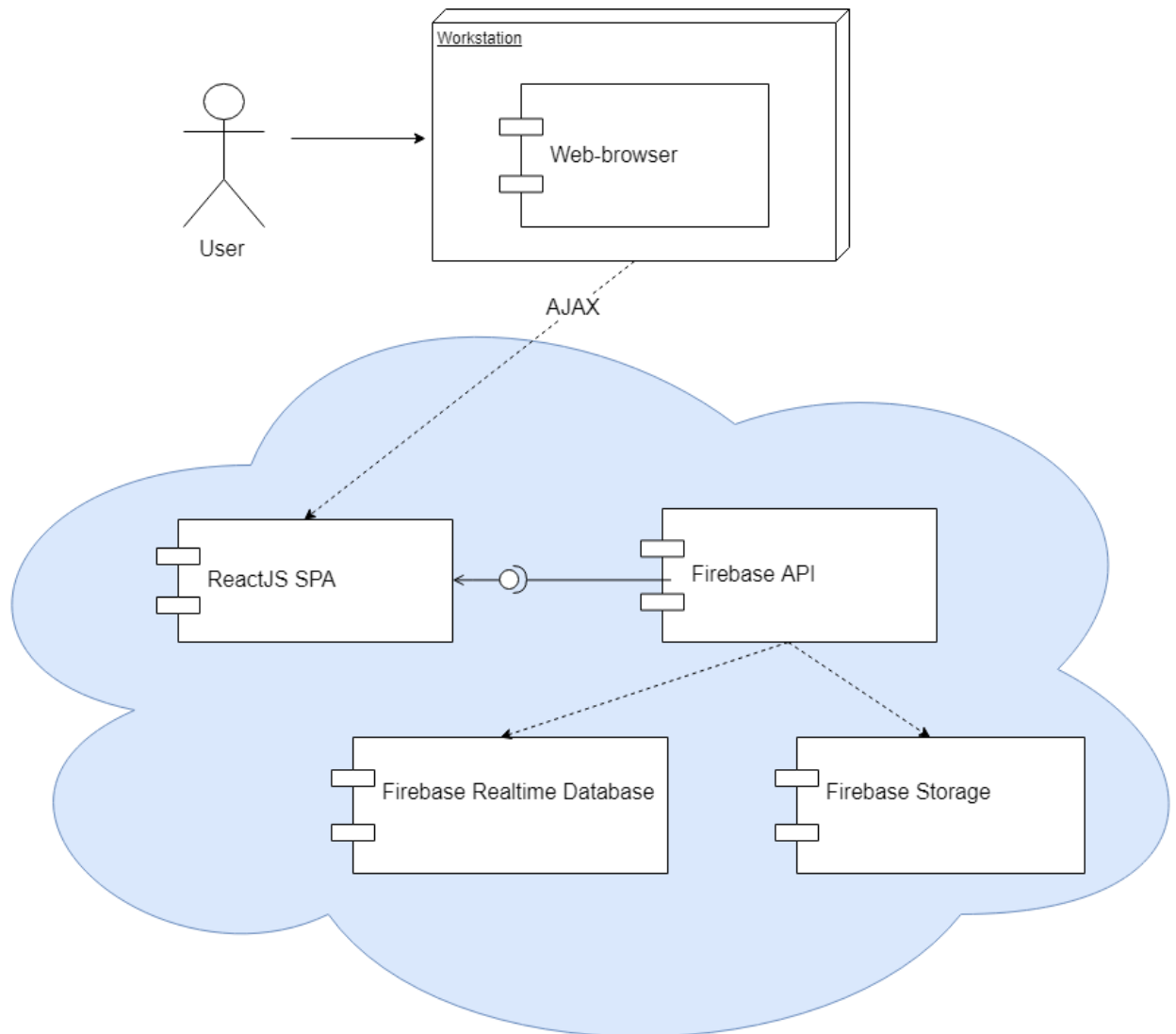
Мета тесту:	Перевірка функції «Отримання рекомендованого керівника»
Початковий стан веб-застосування:	Відкрита сторінка перегляду керівників
Вхідні дані:	Обрати рівень компетентності з Algorithms. Обрати рівень компетентності з Scientific Modeling. Обрати рівень компетентності з Web Technologies. Обрати рівень компетентності з Mobile Technologies. Обрати рівень компетентності з Artificial Intelligence.
Схема проведення тесту:	Обрати рівень компетентності з Algorithms. Обрати рівень компетентності з Scientific Modeling. Обрати рівень компетентності з Web Technologies. Обрати рівень компетентності з Mobile Technologies. Обрати рівень компетентності з Artificial Intelligence. Натиснути кнопку Update.
Очікуваний результат:	Відкрита сторінка перегляду профілів керівників. Рекомендованого керівника підсвічено зеленим.
Стан веб-застосування після проведення вибробувань.	Відкрита сторінка перегляду профілів керівників. Рекомендованого керівника підсвічено зеленим.

Таблиця 6.7 – Тест додавання фото до профілю керівника

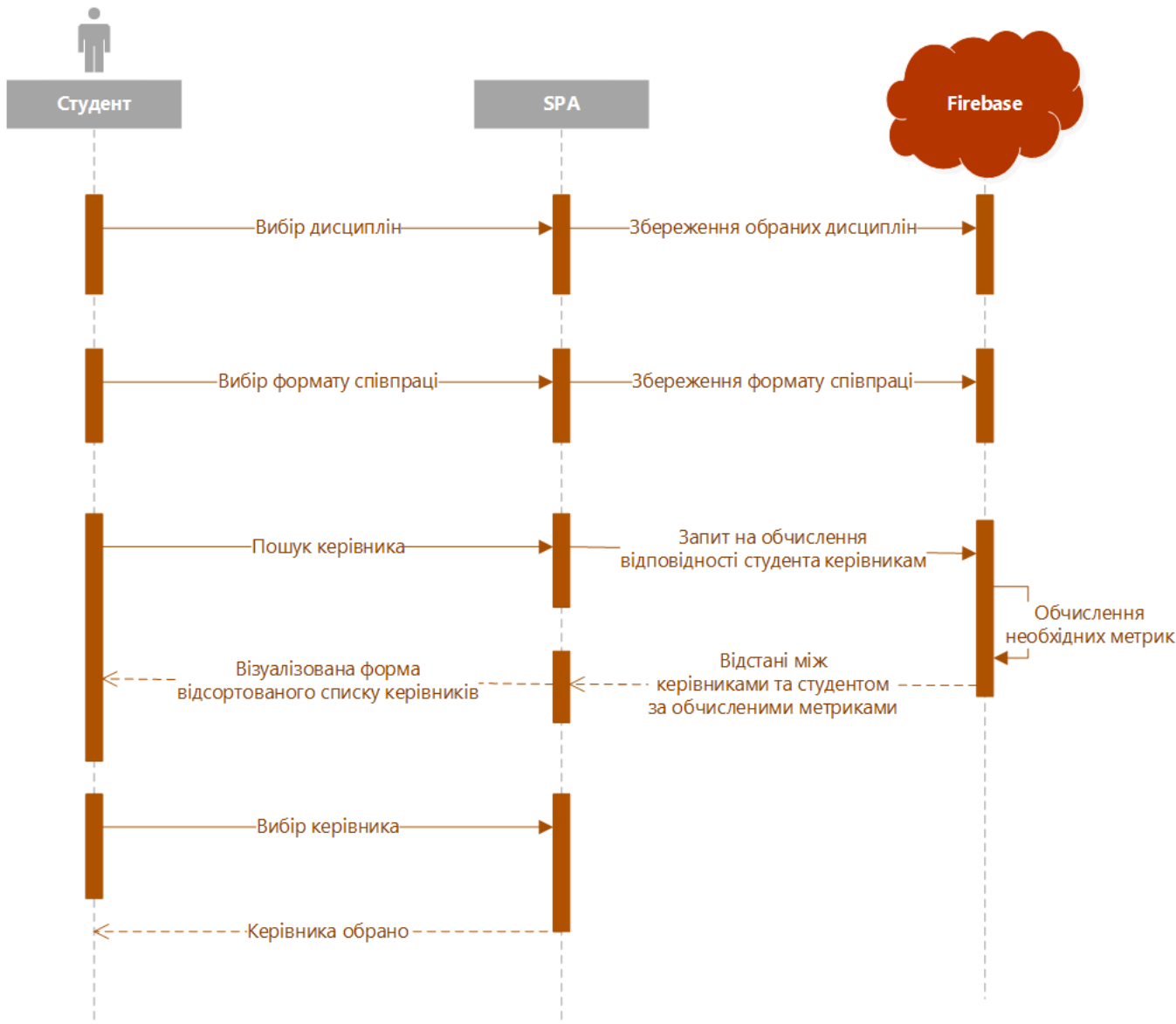
Мета тесту:	Перевірка функції «Додавання фото до профілю керівника»
Початковий стан веб-застосування	Відкрита сторінка внесення інформації про керівника
Вхідні дані:	Фотографія керівника у форматі png, jpeg, gif.
Схема проведення тесту:	Натиснути на кнопку «Обрати файл». Обрати файл форматів png, jpeg, gif. Підтвердити вибір
Очікуваний результат:	Фото керівника додано до профілю і є доступним для перегляду.
Стан веб-застосування після проведення випробувань.	Фото керівника додано до профілю і є доступним для перегляду.



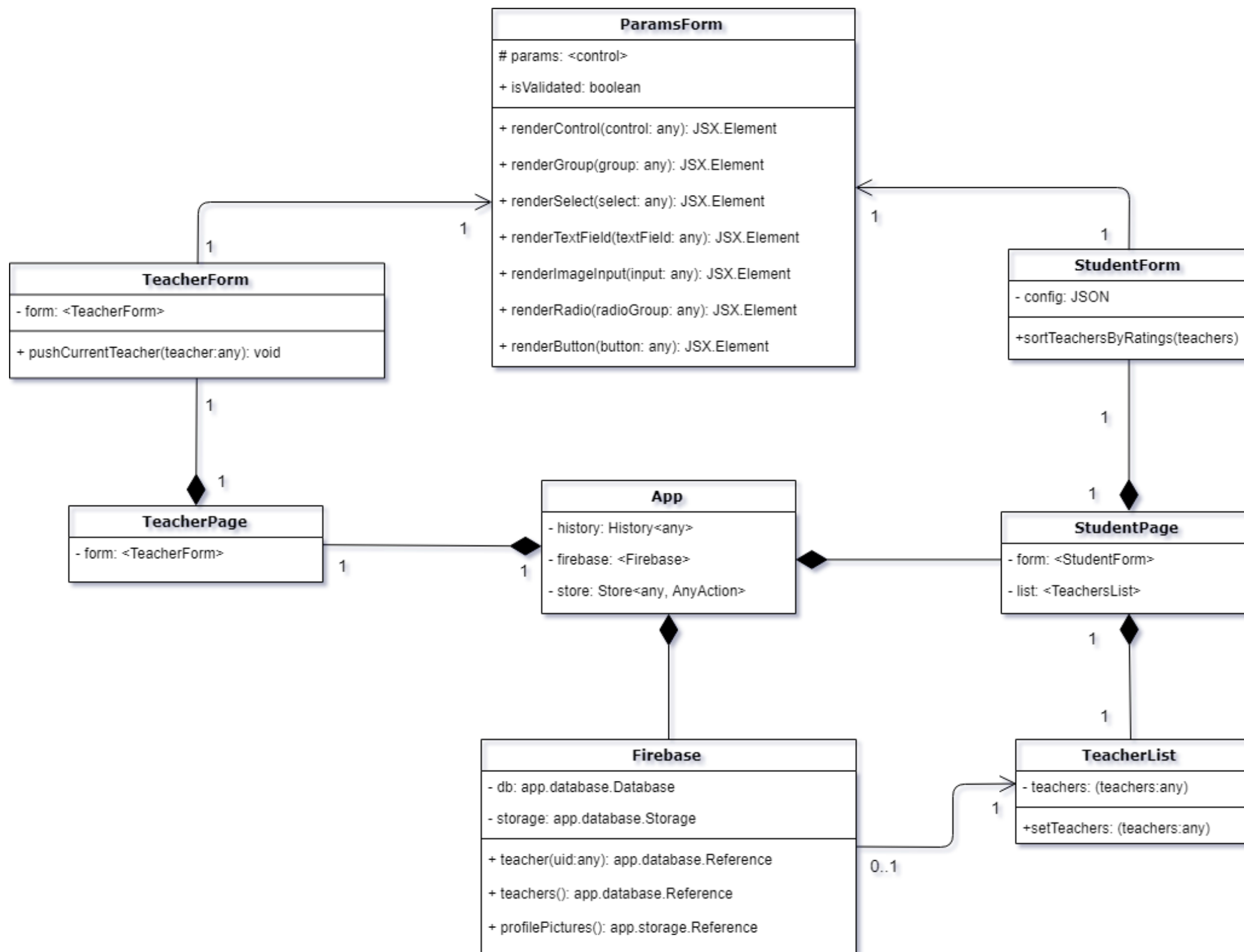
					ДП ІС-5110.1181-с.ССД							
					Схема структурна діяльності	Літера			Маса		Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата								
Розробив		Как С.Р.										
Перевірив		Телишева Т.О.										
Т. кон.						Аркуш 1			Аркушів 1			
					Інформаційна підтримка вибору наукового керівника кваліфікаційної роботи	КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51						
Н. кон.		Телишева Т.О.										
Затвердив		Телишева Т.О.										



					ДП ІС-5110.1181-с.ССК							
					Схема структурна компонентів	Літера			Маса		Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата								
Розробив	Как С.Р.											
Перевірив	Телишева Т.О.											
Т. кон.						Аркуш 1			Аркушів 1			
					Інформаційна підтримка вибору наукового керівника кваліфікаційної роботи	КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51						
Н. кон.	Телишева Т.О.											
Затвердив	Телишева Т.О.											



					ДП ІС-5110.1181-с.ССП							
					Схема структурна послідовності	Літера			Маса		Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата								
Розробив		Как С.Р.										
Перевірів		Телишева Т.О.										
Т. кон.												
					Інформаційна підтримка вибору наукового керівника кваліфікаційної роботи	Аркуш 1			Аркушів 1			
Н. кон.		Телишева Т.О.				КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51						
Затвердив		Телишева Т.О.										



					ДП ІС-5110.1181-с.ССК			
					Схема структурна класів	Літера	Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата				
Розробив	Как С.Р.							
Перевірів	Телишева Т.О.							
Т. кон.					Інформаційна підтримка вибору наукового керівника кваліфікаційної роботи	Аркуш 1		Аркушів 1
Н. кон.	Телишева Т.О.					КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51		
Затвердив	Телишева Т.О.							

Рішення з математичного забезпечення

Дано:

Об'єкт	AL	SM	WT	MT	AI	WF
Керівник 1	$p_{1,1}$	$p_{1,2}$	$p_{1,3}$	$p_{1,4}$	$p_{1,5}$	$p_{1,6}$
Керівник 2	$p_{2,1}$	$p_{2,2}$	$p_{2,3}$	$p_{2,4}$	$p_{2,5}$	$p_{2,6}$
Керівник 3	$p_{3,1}$	$p_{3,2}$	$p_{3,3}$	$p_{3,4}$	$p_{3,5}$	$p_{3,6}$
...
Керівник N	$p_{n,1}$	$p_{n,2}$	$p_{n,3}$	$p_{n,4}$	$p_{n,5}$	$p_{n,6}$
Студент	q_1	q_2	q_3	q_4	q_5	q_6

Зацікавленість і-го керівника у j-му напрямі $p_{i,j}$ ($j = \overline{1,5}$) може набувати значень:

$$p_{i,j} = \begin{cases} 0, \text{ якщо } i - \text{й керівник не цікавиться } j - \text{м напрямом,} \\ 1, \text{ якщо } i - \text{й керівник цікавиться } j - \text{м напрямом,} \\ 2, \text{ якщо } i - \text{й керівник має досвід у } j - \text{му напрямі.} \end{cases} \quad (3.1)$$

Зацікавленість студента у j-му напрямі q_j ($j = \overline{1,5}$) може набувати значень:

$$q_j = \begin{cases} 0, \text{ якщо студент не цікавиться } j - \text{м напрямом,} \\ 1, \text{ якщо студент цікавиться } j - \text{м напрямом,} \\ 2, \text{ якщо студент має досвід у } j - \text{му напрямі.} \end{cases} \quad (3.2)$$

Зручний для і-го керівника формат консультацій $p_{i,6}$ може набувати значень:

$$p_{j,6} = \begin{cases} 0, \text{ якщо керівник обирає } More Distance, \\ 1, \text{ якщо керівник обирає } Equal, \\ 2, \text{ якщо керівник обирає } More Face to Face. \end{cases} \quad (3.3)$$

Зручний для студента формат консультацій q_6 може набувати значень:

$$q_6 = \begin{cases} 0, \text{ якщо студент обирає } More Distance, \\ 1, \text{ якщо студент обирає } Equal, \\ 2, \text{ якщо студент обирає } More Face to Face. \end{cases} \quad (3.4)$$

Необхідно визначити викладача, який найбільше задовольняє побажанням студента в якості керівника дипломного проекту.

Метрика Евкліда:

$$d(p, q)_E = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

Манхеттен метрика:

$$(3.5) \quad d(p, q)_M = \sum_{i=1}^n |p_i - q_i| \quad (3.6)$$

Міра подібності Жаккара

$$d(p, q)_J = 1 - \frac{\sum_{i=1}^n \min(p_i, q_i)}{\sum_{i=1}^n \max(p_i, q_i)} \quad (3.7)$$

Алгоритм розв'язання:

КРОК 1. Визначити зацікавленість студента і керівників науковими напрямками, описаними вище, та сформувати рядки таблиці p та q відповідно.

КРОК 2. Для кожної пари q та p_i розрахувати значення відстані за формулою (3.1) та запам'ятати значення проміжного розрахунку

$$d(p_i, q)_{prom,E} = 100 - (d(p_i, q)_E / \max(d(p, q)_E) * 100) \quad (3.8)$$

де $\max(d(p, q))$ – максимально можлива відстань між q та p_i за обраною метрикою.

КРОК 3. Повторити дії виконані на кроці 2 для розрахунку відстані за формулою (3.6) та запам'ятати значення проміжного розрахунку $d(p_i, q)_{prom,M}$, який розраховується аналогічно до (3.8)

КРОК 4. Для кожної пари q та p_i розрахувати міру подібності за формулою (3.7) та запам'ятати значення проміжного розрахунку $d(p_i, q)_{prom,J}$.

КРОК 5. Відсортувати керівників за спаданням середнього значення отриманих на кроках 1-4 розрахунків. За розробленою концепцією алгоритму перший керівник буде рекомендованим.

Демонстраційний плакат до дипломного проекту

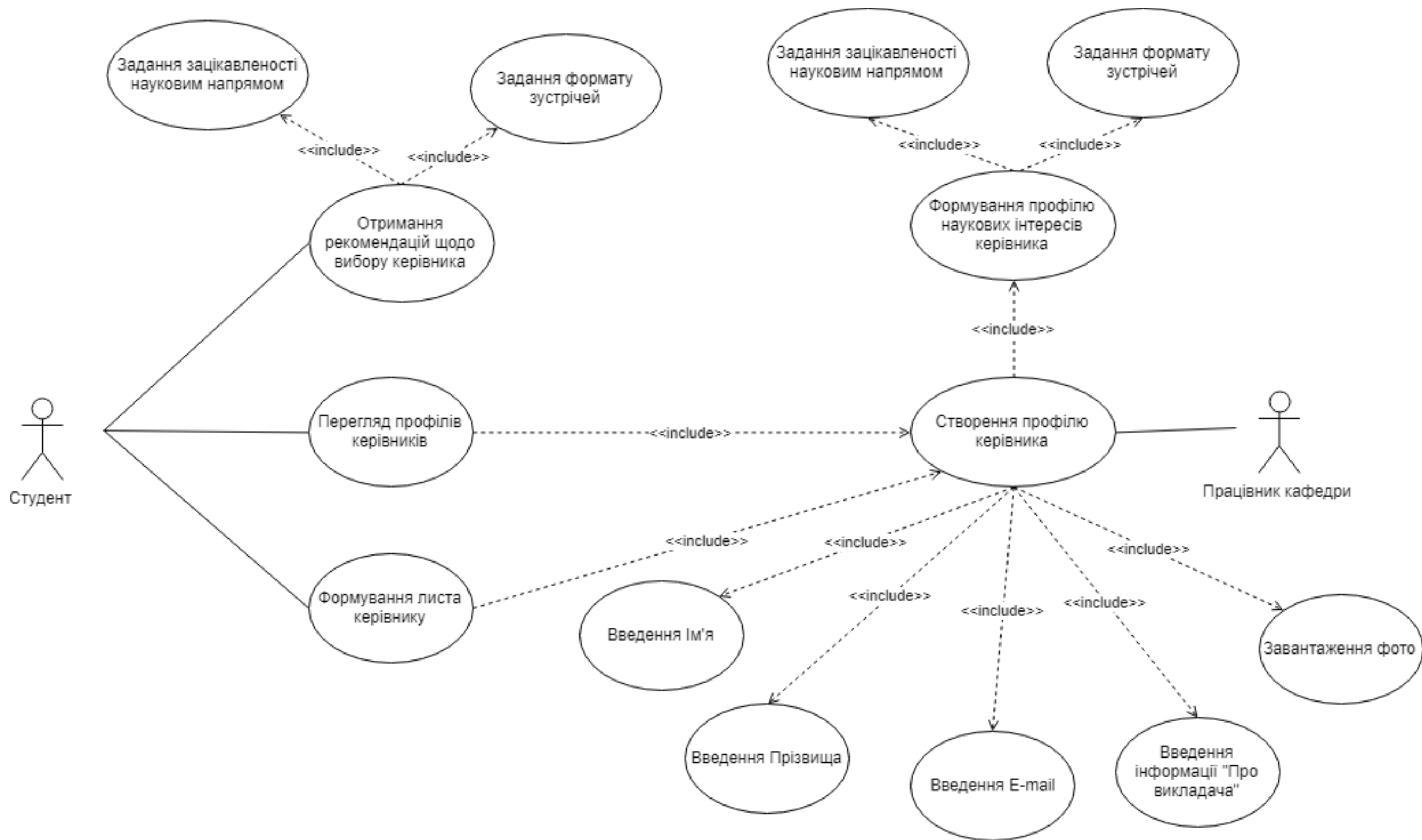
„Інформаційна підтримка вибору наукового керівника кваліфікаційної роботи ”

Виконав студент гр. ІС-51

Как С.Р.

Керівник ДП

Телишева Т.О.



					ДП ІС-5110.1181-с.ССВ			
					Схема структурна варіантів використання	Літера	Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата				
Розробив	Как С.Р.							
Перевірів	Телишева Т.О.							
Т. кон.					Інформаційна підтримка вибору наукового керівника кваліфікаційної роботи	Аркуш 1		Аркушів 1
Н. кон.	Телишева Т.О.					КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51		
Затвердив	Телишева Т.О.							

Сторінка перегляду керівників

General Information:

Name

Surname

Details about you

E-mail

Выберите файл

Файл ...бран

Interests:

Algorithms

Not interested

Scientific Modeling

Not interested

Web Technologies

Not interested

Mobile Technologies

Not interested

Artificial Intelligence

Not interested

Working format:

☒ More distance

☐ Equal

☐ More Face to Face

ADD

Форма внесення інформації про керівника

Interests:

Algorithms
Experienced

Scientific Modeling
Not interested

Web Technologies
Interested

Mobile Technologies
Experienced

Artificial Intelligence
Interested


Working format:

☐ More distance


☐ Equal

☒ More Face to Face


UPDATE



Тарас Клименок
Кандидат наук. Маю практичний досвід та глибокі знання у розробці мобільних архітектур.
E-mail: tklim@meta.ua
Interests:
◦ web
◦ mobile
Chooosed: 1 times.



Дмитро Чапля
Старший викладач. Маю практичний досвід розробки мобільних застосунків.
E-mail: dchaplya@ukr.net
Interests:
◦ algorithms
◦ mobile
Chooosed: times.



Сергій Іванов
Старший викладач. Маю досвід практичної роботи з системами штучного інтелекту.
E-mail: sivanov@gmail.com
Interests:
◦ algorithms
◦ ai
Chooosed: times.

Similarities:
 Euclidian :
 66.830 %
 Manhattan :
 77.778 %
 Jaccard :
 42.857 %

					ДП ІС-5110.1181-с.КЕ						
						<i>Креслення вигляду екранних форм</i>	Літера			Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата							
Розробив		Как С.Р.									
Перевірів		Телишева Т.О.									
Т. кон.						Аркуш 1			Аркушів 1		
					<i>Інформаційна підтримка вибору наукового керівника кваліфікаційної роботи</i>	<i>КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51</i>					
Н. кон.		Телишева Т.О.									
Затвердив		Телишева Т.О.									